

Cisco Security Appliance Command Line Configuration Guide

For the Cisco ASA 5500 Series and Cisco PIX 500 Series

Software Version 8.0

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Customer Order Number: N/A, Online only
Text Part Number: OL-12172-04

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

CCDE, CCENT, CCSI, Cisco Eos, Cisco HealthPresence, Cisco IronPort, the Cisco logo, Cisco Nurse Connect, Cisco Pulse, Cisco SensorBase, Cisco StackPower, Cisco StadiumVision, Cisco TelePresence, Cisco Unified Computing System, Cisco WebEx, DCE, Flip Channels, Flip for Good, Flip Mino, Flipshare (Design), Flip Ultra, Flip Video, Flip Video (Design), Instant Broadband, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn, Cisco Capital, Cisco Capital (Design), Cisco:Financed (Stylized), Cisco Store, Flip Gift Card, and One Million Acts of Green are service marks; and Access Registrar, Aironet, AllTouch, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Lumin, Cisco Nexus, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, Continuum, EtherFast, EtherSwitch, Event Center, Explorer, Follow Me Browsing, GainMaker, iLYNX, IOS, iPhone, IronPort, the IronPort logo, Laser Link, LightStream, Linksys, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, PCNow, PIX, PowerKEY, PowerPanels, PowerTV, PowerTV (Design), PowerVu, Prisma, ProConnect, ROSA, SenderBase, SMARTnet, Spectrum Expert, StackWise, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0910R)

Cisco Security Appliance Command Line Configuration Guide
Copyright © 2009 Cisco Systems, Inc. All rights reserved.



CONTENTS

About This Guide xli

Document Objectives xli

Audience xli

Related Documentation xlii

Document Conventions xlii

Obtaining Documentation, Obtaining Support, and Security Guidelines xlii

PART 1 **Getting Started and General Information**

CHAPTER 1 **Introduction to the Security Appliance** 1-1

Supported Platform Models 1-1

SSM and SSC Support Per Model 1-2

VPN Specifications 1-3

New Features 1-3

New Features in Version 8.0(5) 1-3

New Features in Version 8.0(4) 1-4

New Features in Version 8.0(3) 1-8

New Features in Version 8.0(2) 1-9

Firewall Functional Overview 1-14

Security Policy Overview 1-15

Permitting or Denying Traffic with Access Lists 1-15

Applying NAT 1-15

Protecting from IP Fragments 1-15

Using AAA for Through Traffic 1-15

Applying HTTP, HTTPS, or FTP Filtering 1-16

Applying Application Inspection 1-16

Sending Traffic to the Advanced Inspection and Prevention Security Services Module 1-16

Sending Traffic to the Content Security and Control Security Services Module 1-16

Applying QoS Policies 1-16

Applying Connection Limits and TCP Normalization 1-16

Enabling Threat Detection 1-17

Firewall Mode Overview 1-17

Stateful Inspection Overview 1-17

VPN Functional Overview 1-18

Security Context Overview 1-19

CHAPTER 2 **Getting Started** 2-1

Getting Started with Your Platform Model 2-1

Factory Default Configurations 2-1

Restoring the Factory Default Configuration 2-2

ASA 5505 Default Configuration 2-2

ASA 5510 and Higher Default Configuration 2-3

PIX 515/515E Default Configuration 2-4

Accessing the Command-Line Interface 2-4

Setting Transparent or Routed Firewall Mode 2-5

Working with the Configuration 2-6

Saving Configuration Changes 2-6

Saving Configuration Changes in Single Context Mode 2-7

Saving Configuration Changes in Multiple Context Mode 2-7

Copying the Startup Configuration to the Running Configuration 2-8

Viewing the Configuration 2-8

Clearing and Removing Configuration Settings 2-9

Creating Text Configuration Files Offline 2-9

CHAPTER 3 **Managing Feature Licenses** 3-1

Supported Feature Licenses Per Model 3-1

Information About Feature Licenses 3-9

Preinstalled License 3-10

VPN Flex and Evaluation Licenses 3-10

How the Temporary License Timer Works 3-10

How Multiple Licenses Interact 3-11

Failover and Temporary Licenses 3-11

Guidelines and Limitations 3-12

Viewing Your Current License 3-12

Obtaining an Activation Key 3-14

Entering a New Activation Key 3-15

Upgrading the License for a Failover Pair 3-16

Upgrading the License for a Failover (No Reload Required) 3-16

Upgrading the License for a Failover (Reload Required) 3-17

Feature History for Licensing 3-18

CHAPTER 4 **Enabling Multiple Context Mode** 4-1

Security Context Overview 4-1
Common Uses for Security Contexts 4-2
Unsupported Features 4-2
Context Configuration Files 4-2
Context Configurations 4-2
System Configuration 4-3
Admin Context Configuration 4-3
How the Security Appliance Classifies Packets 4-3
Valid Classifier Criteria 4-3
Invalid Classifier Criteria 4-4
Classification Examples 4-5
Cascading Security Contexts 4-8
Management Access to Security Contexts 4-9
System Administrator Access 4-9
Context Administrator Access 4-10
Enabling or Disabling Multiple Context Mode 4-10
Backing Up the Single Mode Configuration 4-10
Enabling Multiple Context Mode 4-10
Restoring Single Context Mode 4-11

CHAPTER 5 **Configuring Switch Ports and VLAN Interfaces for the Cisco ASA 5505 Adaptive Security Appliance** 5-1

Interface Overview 5-1
Understanding ASA 5505 Ports and Interfaces 5-2
Maximum Active VLAN Interfaces for Your License 5-2
Default Interface Configuration 5-4
VLAN MAC Addresses 5-4
Power Over Ethernet 5-4
Monitoring Traffic Using SPAN 5-4
Security Level Overview 5-5
Configuring VLAN Interfaces 5-5
Configuring Switch Ports as Access Ports 5-9
Configuring a Switch Port as a Trunk Port 5-11
Allowing Communication Between VLAN Interfaces on the Same Security Level 5-13

CHAPTER 6 **Configuring Ethernet Settings, Redundant Interfaces, and Subinterfaces** 6-1

Configuring and Enabling RJ-45 Interfaces 6-1
RJ-45 Interface Overview 6-2

Default State of Physical Interfaces 6-2
Connector Types 6-2
Auto-MDI/MDIX Feature 6-2
Configuring the RJ-45 Interface 6-2
Configuring and Enabling Fiber Interfaces 6-3
Default State of Physical Interfaces 6-3
Configuring the Fiber Interface 6-4
Configuring a Redundant Interface 6-4
Redundant Interface Overview 6-5
Default State of Redundant Interfaces 6-5
Redundant Interfaces and Failover Guidelines 6-5
Redundant Interface MAC Address 6-5
Physical Interface Guidelines 6-5
Adding a Redundant Interface 6-6
Changing the Active Interface 6-7
Configuring VLAN Subinterfaces and 802.1Q Trunking 6-7
Subinterface Overview 6-7
Default State of Subinterfaces 6-7
Maximum Subinterfaces 6-8
Preventing Untagged Packets on the Physical Interface 6-8
Adding a Subinterface 6-8

CHAPTER 7 **Adding and Managing Security Contexts** 7-1
Configuring Resource Management 7-1
Classes and Class Members Overview 7-1
Resource Limits 7-2
Default Class 7-3
Class Members 7-4
Configuring a Class 7-4
Configuring a Security Context 7-7
Automatically Assigning MAC Addresses to Context Interfaces 7-11
Information About MAC Addresses 7-11
Default MAC Address 7-11
Interaction with Manual MAC Addresses 7-11
Failover MAC Addresses 7-12
MAC Address Format 7-12
Enabling Auto-Generation of MAC Addresses 7-12
Viewing Assigned MAC Addresses 7-13
Viewing MAC Addresses in the System Configuration 7-13

Viewing MAC Addresses Within a Context 7-14
Changing Between Contexts and the System Execution Space 7-14
Managing Security Contexts 7-15
Removing a Security Context 7-15
Changing the Admin Context 7-16
Changing the Security Context URL 7-16
Reloading a Security Context 7-17
Reloading by Clearing the Configuration 7-17
Reloading by Removing and Re-adding the Context 7-18
Monitoring Security Contexts 7-18
Viewing Context Information 7-18
Viewing Resource Allocation 7-19
Viewing Resource Usage 7-22
Monitoring SYN Attacks in Contexts 7-23

CHAPTER 8 **Configuring Interface Parameters** 8-1
Security Level Overview 8-1
Configuring Interface Parameters 8-2
Interface Parameters Overview 8-2
Default State of Interfaces 8-3
Default Security Level 8-3
Multiple Context Mode Guidelines 8-3
Configuring the Interface 8-3
Allowing Communication Between Interfaces on the Same Security Level 8-7

CHAPTER 9 **Configuring Basic Settings** 9-1
Changing the Login Password 9-1
Changing the Enable Password 9-1
Setting the Hostname 9-2
Setting the Domain Name 9-2
Setting the Date and Time 9-2
Setting the Time Zone and Daylight Saving Time Date Range 9-3
Setting the Date and Time Using an NTP Server 9-4
Setting the Date and Time Manually 9-4
Setting the Management IP Address for a Transparent Firewall 9-5

CHAPTER 10 **Configuring IP Routing** 10-1
How Routing Behaves Within the ASA Security Appliance 10-1

Egress Interface Selection Process 10-1
Next Hop Selection Process 10-2
Configuring Static and Default Routes 10-2
Configuring a Static Route 10-3
Configuring a Default Static Route 10-4
Configuring Static Route Tracking 10-5
Defining Route Maps 10-7
Configuring OSPF 10-8
OSPF Overview 10-9
Enabling OSPF 10-10
Redistributing Routes Into OSPF 10-10
Configuring OSPF Interface Parameters 10-12
Configuring OSPF Area Parameters 10-14
Configuring OSPF NSSA 10-15
Configuring Route Summarization Between OSPF Areas 10-16
Configuring Route Summarization When Redistributing Routes into OSPF 10-16
Defining Static OSPF Neighbors 10-17
Generating a Default Route 10-17
Configuring Route Calculation Timers 10-18
Logging Neighbors Going Up or Down 10-18
Displaying OSPF Update Packet Pacing 10-19
Monitoring OSPF 10-19
Restarting the OSPF Process 10-20
Configuring RIP 10-20
Enabling and Configuring RIP 10-21
Redistributing Routes into the RIP Routing Process 10-22
Configuring RIP Send/Receive Version on an Interface 10-23
Enabling RIP Authentication 10-23
Monitoring RIP 10-24
Configuring EIGRP 10-24
EIGRP Routing Overview 10-25
Enabling and Configuring EIGRP Routing 10-26
Enabling and Configuring EIGRP Stub Routing 10-27
Enabling EIGRP Authentication 10-27
Defining an EIGRP Neighbor 10-28
Redistributing Routes Into EIGRP 10-29
Configuring the EIGRP Hello Interval and Hold Time 10-30
Disabling Automatic Route Summarization 10-30
Configuring Summary Aggregate Addresses 10-31

Disabling EIGRP Split Horizon 10-31
Changing the Interface Delay Value 10-32
Monitoring EIGRP 10-32
Disabling Neighbor Change and Warning Message Logging 10-32
The Routing Table 10-33
Displaying the Routing Table 10-33
How the Routing Table is Populated 10-33
Backup Routes 10-35
How Forwarding Decisions are Made 10-35
Dynamic Routing and Failover 10-36

CHAPTER 11 **Configuring DHCP, DDNS, and WCCP Services** 11-1
Configuring a DHCP Server 11-1
Enabling the DHCP Server 11-2
Configuring DHCP Options 11-3
Using Cisco IP Phones with a DHCP Server 11-4
Configuring DHCP Relay Services 11-5
Configuring Dynamic DNS 11-6
Example 1: Client Updates Both A and PTR RRs for Static IP Addresses 11-7
Example 2: Client Updates Both A and PTR RRs; DHCP Server Honors Client Update Request; FQDN Provided Through Configuration 11-7
Example 3: Client Includes FQDN Option Instructing Server Not to Update Either RR; Server Overrides Client and Updates Both RRs. 11-8
Example 4: Client Asks Server To Perform Both Updates; Server Configured to Update PTR RR Only; Honors Client Request and Updates Both A and PTR RR 11-9
Example 5: Client Updates A RR; Server Updates PTR RR 11-9
Configuring Web Cache Services Using WCCP 11-9
WCCP Feature Support 11-10
WCCP Interaction With Other Features 11-10
Enabling WCCP Redirection 11-11

CHAPTER 12 **Configuring Multicast Routing** 12-1
Multicast Routing Overview 12-1
Enabling Multicast Routing 12-2
Configuring IGMP Features 12-2
Disabling IGMP on an Interface 12-3
Configuring Group Membership 12-3
Configuring a Statically Joined Group 12-3
Controlling Access to Multicast Groups 12-3

Limiting the Number of IGMP States on an Interface 12-4
Modifying the Query Interval and Query Timeout 12-4
Changing the Query Response Time 12-5
Changing the IGMP Version 12-5
Configuring Stub Multicast Routing 12-5
Configuring a Static Multicast Route 12-6
Configuring PIM Features 12-6
Disabling PIM on an Interface 12-6
Configuring a Static Rendezvous Point Address 12-7
Configuring the Designated Router Priority 12-7
Filtering PIM Register Messages 12-7
Configuring PIM Message Intervals 12-8
Configuring a Multicast Boundary 12-8
Filtering PIM Neighbors 12-8
Supporting Mixed Bidirectional/Sparse-Mode PIM Networks 12-9
For More Information about Multicast Routing 12-10

CHAPTER 13 **Configuring IPv6** 13-1
IPv6-enabled Commands 13-1
Configuring IPv6 13-2
Configuring IPv6 on an Interface 13-3
Configuring a Dual IP Stack on an Interface 13-4
Enforcing the Use of Modified EUI-64 Interface IDs in IPv6 Addresses 13-4
Configuring IPv6 Duplicate Address Detection 13-4
Configuring IPv6 Default and Static Routes 13-5
Configuring IPv6 Access Lists 13-6
Configuring IPv6 Neighbor Discovery 13-7
Configuring Neighbor Solicitation Messages 13-7
Configuring Router Advertisement Messages 13-9
Configuring a Static IPv6 Neighbor 13-11
Verifying the IPv6 Configuration 13-11
The show ipv6 interface Command 13-11
The show ipv6 route Command 13-12

CHAPTER 14 **Configuring AAA Servers and the Local Database** 14-1
AAA Overview 14-1
About Authentication 14-2
About Authorization 14-2
About Accounting 14-2

AAA Server and Local Database Support 14-3
Summary of Support 14-3
RADIUS Server Support 14-4
Authentication Methods 14-4
Attribute Support 14-4
RADIUS Authorization Functions 14-5
TACACS+ Server Support 14-5
RSA/SDI Server Support 14-5
RSA/SDI Version Support 14-5
Two-step Authentication Process 14-5
SDI Primary and Replica Servers 14-5
NT Server Support 14-6
Kerberos Server Support 14-6
LDAP Server Support 14-6
SSO Support for Clientless SSL VPN with HTTP Forms 14-6
Local Database Support 14-6
User Profiles 14-7
Fallback Support 14-7
Configuring the Local Database 14-7
Identifying AAA Server Groups and Servers 14-9
Configuring an LDAP Server 14-12
Authentication with LDAP 14-13
Authorization with LDAP for VPN 14-14
LDAP Attribute Mapping 14-15
Using Certificates and User Login Credentials 14-16
Using User Login Credentials 14-16
Using certificates 14-17
Supporting a Zone Labs Integrity Server 14-17
Overview of Integrity Server and Security Appliance Interaction 14-18
Configuring Integrity Server Support 14-18

CHAPTER 15 **Configuring Failover** 15-1
Understanding Failover 15-1
Failover System Requirements 15-2
Hardware Requirements 15-2
Software Requirements 15-2
License Requirements 15-3
The Failover and Stateful Failover Links 15-3
Failover Link 15-3

Stateful Failover Link 15-5
Active/Active and Active/Standby Failover 15-6
Active/Standby Failover 15-7
Active/Active Failover 15-11
Determining Which Type of Failover to Use 15-15
Stateless (Regular) and Stateful Failover 15-16
Stateless (Regular) Failover 15-16
Stateful Failover 15-16
Failover Health Monitoring 15-18
Unit Health Monitoring 15-18
Interface Monitoring 15-18
Failover Feature/Platform Matrix 15-19
Failover Times by Platform 15-20
Configuring Failover 15-20
Failover Configuration Limitations 15-20
Configuring Active/Standby Failover 15-21
Prerequisites 15-21
Configuring Cable-Based Active/Standby Failover (PIX 500 Series Security Appliance Only) 15-21
Configuring LAN-Based Active/Standby Failover 15-23
Configuring Optional Active/Standby Failover Settings 15-26
Configuring Active/Active Failover 15-29
Prerequisites 15-29
Configuring Cable-Based Active/Active Failover (PIX 500 series security appliance) 15-29
Configuring LAN-Based Active/Active Failover 15-31
Configuring Optional Active/Active Failover Settings 15-35
Configuring Unit Health Monitoring 15-41
Configuring Failover Communication Authentication/Encryption 15-41
Verifying the Failover Configuration 15-42
Using the show failover Command 15-42
Viewing Monitored Interfaces 15-50
Displaying the Failover Commands in the Running Configuration 15-50
Testing the Failover Functionality 15-51
Controlling and Monitoring Failover 15-51
Forcing Failover 15-51
Disabling Failover 15-52
Restoring a Failed Unit or Failover Group 15-52
Monitoring Failover 15-53
Failover System Messages 15-53
Debug Messages 15-53

SNMP 15-53
Remote Command Execution 15-53
Changing Command Modes 15-54
Security Considerations 15-55
Limitations of Remote Command Execution 15-55
Auto Update Server Support in Failover Configurations 15-56
Auto Update Process Overview 15-56
Monitoring the Auto Update Process 15-57

CHAPTER 16 **Using Modular Policy Framework** 16-1
Information About Modular Policy Framework 16-1
Modular Policy Framework Supported Features 16-1
Modular Policy Framework Configuration Overview 16-2
Default Global Policy 16-3
Identifying Traffic (Layer 3/4 Class Map) 16-4
Default Class Maps 16-4
Maximum Class Maps 16-5
Creating a Layer 3/4 Class Map for Through Traffic 16-5
Creating a Layer 3/4 Class Map for Management Traffic 16-7
Configuring Special Actions for Application Inspections (Inspection Policy Map) 16-8
Inspection Policy Map Overview 16-9
Defining Actions in an Inspection Policy Map 16-9
Identifying Traffic in an Inspection Class Map 16-12
Creating a Regular Expression 16-13
Creating a Regular Expression Class Map 16-16
Defining Actions (Layer 3/4 Policy Map) 16-16
Information About Layer 3/4 Policy Maps 16-17
Policy Map Guidelines 16-17
Hierarchical Policy Maps 16-17
Feature Directionality 16-18
Feature Matching Guidelines Within a Policy Map 16-18
Order in Which Multiple Feature Actions are Applied 16-19
Incompatibility of Certain Feature Actions 16-20
Feature Matching Guidelines for Multiple Policy Maps 16-21
Default Layer 3/4 Policy Map 16-21
Adding a Layer 3/4 Policy Map 16-22
Applying Actions to an Interface (Service Policy) 16-23
Modular Policy Framework Examples 16-24
Applying Inspection and QoS Policing to HTTP Traffic 16-25

Applying Inspection to HTTP Traffic Globally 16-25
Applying Inspection and Connection Limits to HTTP Traffic to Specific Servers 16-26
Applying Inspection to HTTP Traffic with NAT 16-27

PART 2 Configuring the Firewall

CHAPTER 17 Firewall Mode Overview 17-1
Routed Mode Overview 17-1
IP Routing Support 17-1
How Data Moves Through the Security Appliance in Routed Firewall Mode 17-1
An Inside User Visits a Web Server 17-2
An Outside User Visits a Web Server on the DMZ 17-3
An Inside User Visits a Web Server on the DMZ 17-4
An Outside User Attempts to Access an Inside Host 17-5
A DMZ User Attempts to Access an Inside Host 17-6
Transparent Mode Overview 17-7
Transparent Firewall Network 17-7
Allowing Layer 3 Traffic 17-7
Allowed MAC Addresses 17-7
Passing Traffic Not Allowed in Routed Mode 17-8
MAC Address vs. Route Lookups 17-8
Using the Transparent Firewall in Your Network 17-9
Transparent Firewall Guidelines 17-9
Unsupported Features in Transparent Mode 17-10
How Data Moves Through the Transparent Firewall 17-11
An Inside User Visits a Web Server 17-12
An Inside User Visits a Web Server Using NAT 17-13
An Outside User Visits a Web Server on the Inside Network 17-14
An Outside User Attempts to Access an Inside Host 17-15

CHAPTER 18 Identifying Traffic with Access Lists 18-1
Access List Overview 18-1
Access List Types 18-2
Access Control Entry Order 18-2
Access Control Implicit Deny 18-3
IP Addresses Used for Access Lists When You Use NAT 18-3
Adding an Extended Access List 18-5
Extended Access List Overview 18-5
Allowing Broadcast and Multicast Traffic through the Transparent Firewall 18-6

Adding an Extended ACE 18-7

Adding an EtherType Access List 18-8

EtherType Access List Overview 18-8

Supported EtherTypes 18-9

Implicit Permit of IP and ARPs Only 18-9

Implicit and Explicit Deny ACE at the End of an Access List 18-9

IPv6 Unsupported 18-9

Using Extended and EtherType Access Lists on the Same Interface 18-9

Allowing MPLS 18-10

Adding an EtherType ACE 18-10

Adding a Standard Access List 18-11

Adding a Webtype Access List 18-11

Simplifying Access Lists with Object Grouping 18-12

How Object Grouping Works 18-13

Adding Object Groups 18-13

Adding a Protocol Object Group 18-14

Adding a Network Object Group 18-14

Adding a Service Object Group 18-15

Adding an ICMP Type Object Group 18-16

Nesting Object Groups 18-16

Using Object Groups with an Access List 18-17

Displaying Object Groups 18-18

Removing Object Groups 18-19

Adding Remarks to Access Lists 18-19

Scheduling Extended Access List Activation 18-19

Adding a Time Range 18-19

Applying the Time Range to an ACE 18-20

Logging Access List Activity 18-21

Access List Logging Overview 18-21

Configuring Logging for an Access Control Entry 18-22

Managing Deny Flows 18-23

CHAPTER 19 **Configuring NAT** 19-1

NAT Overview 19-1

Introduction to NAT 19-1

NAT in Routed Mode 19-2

NAT in Transparent Mode 19-3

NAT Control 19-5

NAT Types 19-6

Dynamic NAT 19-6
PAT 19-8
Static NAT 19-9
Static PAT 19-9
Bypassing NAT When NAT Control is Enabled 19-10
Policy NAT 19-11
NAT and Same Security Level Interfaces 19-15
Order of NAT Commands Used to Match Real Addresses 19-16
Mapped Address Guidelines 19-16
DNS and NAT 19-17
Configuring NAT Control 19-18
Using Dynamic NAT and PAT 19-19
Dynamic NAT and PAT Implementation 19-19
Configuring Dynamic NAT or PAT 19-25
Using Static NAT 19-28
Using Static PAT 19-29
Bypassing NAT 19-32
Configuring Identity NAT 19-32
Configuring Static Identity NAT 19-33
Configuring NAT Exemption 19-35
NAT Examples 19-36
Overlapping Networks 19-36
Redirecting Ports 19-38

CHAPTER 20 **Permitting or Denying Network Access** 20-1
Inbound and Outbound Access List Overview 20-1
Applying an Access List to an Interface 20-2

CHAPTER 21 **Applying AAA for Network Access** 21-1
AAA Performance 21-1
Configuring Authentication for Network Access 21-1
Authentication Overview 21-2
One-Time Authentication 21-2
Applications Required to Receive an Authentication Challenge 21-2
Security Appliance Authentication Prompts 21-2
Static PAT and HTTP 21-3
Enabling Network Access Authentication 21-3
Enabling Secure Authentication of Web Clients 21-5

Authenticating Directly with the Security Appliance 21-6
Enabling Direct Authentication Using HTTP and HTTPS 21-6
Enabling Direct Authentication Using Telnet 21-7
Configuring Authorization for Network Access 21-8
Configuring TACACS+ Authorization 21-8
Configuring RADIUS Authorization 21-10
Configuring a RADIUS Server to Send Downloadable Access Control Lists 21-10
Configuring a RADIUS Server to Download Per-User Access Control List Names 21-14
Configuring Accounting for Network Access 21-14
Using MAC Addresses to Exempt Traffic from Authentication and Authorization 21-16

CHAPTER 22 **Applying Filtering Services** 22-1
Filtering Overview 22-1
Filtering ActiveX Objects 22-2
ActiveX Filtering Overview 22-2
Enabling ActiveX Filtering 22-2
Filtering Java Applets 22-3
Filtering URLs and FTP Requests with an External Server 22-4
URL Filtering Overview 22-4
Identifying the Filtering Server 22-4
Buffering the Content Server Response 22-6
Caching Server Addresses 22-6
Filtering HTTP URLs 22-7
Configuring HTTP Filtering 22-7
Enabling Filtering of Long HTTP URLs 22-7
Truncating Long HTTP URLs 22-7
Exempting Traffic from Filtering 22-8
Filtering HTTPS URLs 22-8
Filtering FTP Requests 22-9
Viewing Filtering Statistics and Configuration 22-9
Viewing Filtering Server Statistics 22-10
Viewing Buffer Configuration and Statistics 22-11
Viewing Caching Statistics 22-11
Viewing Filtering Performance Statistics 22-11
Viewing Filtering Configuration 22-12

CHAPTER 23 **Managing the AIP SSM and CSC SSM** 23-1
Managing the AIP SSM 23-1

AIP SSM Overview 23-1
How the AIP SSM Works with the Adaptive Security Appliance 23-2
Operating Modes 23-3
Using Virtual Sensors 23-3
AIP SSM Procedure Overview 23-4
Sessioning to the AIP SSM 23-5
Configuring the Security Policy on the AIP SSM 23-6
Assigning Virtual Sensors to Security Contexts 23-6
Diverting Traffic to the AIP SSM 23-8
Managing the CSC SSM 23-9
About the CSC SSM 23-10
Getting Started with the CSC SSM 23-12
Determining What Traffic to Scan 23-13
Limiting Connections Through the CSC SSM 23-15
Diverting Traffic to the CSC SSM 23-16
Checking SSM Status 23-18
Transferring an Image onto an SSM 23-19

CHAPTER 24 **Preventing Network Attacks** 24-1
Configuring Threat Detection 24-1
Configuring Basic Threat Detection 24-1
Basic Threat Detection Overview 24-2
Configuring Basic Threat Detection 24-2
Managing Basic Threat Statistics 24-4
Configuring Scanning Threat Detection 24-5
Enabling Scanning Threat Detection 24-5
Managing Shunned Hosts 24-6
Viewing Attackers and Targets 24-7
Configuring and Viewing Threat Statistics 24-7
Configuring Threat Statistics 24-7
Viewing Threat Statistics 24-8
Configuring TCP Normalization 24-12
TCP Normalization Overview 24-12
Enabling the TCP Normalizer 24-12
Configuring Connection Limits and Timeouts 24-17
Connection Limit Overview 24-17
TCP Intercept Overview 24-18
Disabling TCP Intercept for Management Packets for Clientless SSL Compatibility 24-18
Dead Connection Detection (DCD) Overview 24-18

TCP Sequence Randomization Overview 24-18
Enabling Connection Limits and Timeouts 24-19
Preventing IP Spoofing 24-21
Configuring the Fragment Size 24-22
Blocking Unwanted Connections 24-22
Configuring IP Audit for Basic IPS Support 24-23

CHAPTER 25 **Configuring QoS** 25-1
QoS Overview 25-1
Supported QoS Features 25-2
What is a Token Bucket? 25-2
Policing Overview 25-3
Priority Queueing Overview 25-3
Traffic Shaping Overview 25-4
How QoS Features Interact 25-4
DSCP and DiffServ Preservation 25-5
Creating the Standard Priority Queue for an Interface 25-5
Determining the Queue and TX Ring Limits 25-6
Configuring the Priority Queue 25-7
Identifying Traffic for QoS Using Class Maps 25-8
Creating a QoS Class Map 25-8
QoS Class Map Examples 25-8
Creating a Policy for Standard Priority Queueing and/or Policing 25-9
Creating a Policy for Traffic Shaping and Hierarchical Priority Queueing 25-11
Viewing QoS Statistics 25-13
Viewing QoS Police Statistics 25-13
Viewing QoS Standard Priority Statistics 25-14
Viewing QoS Shaping Statistics 25-14
Viewing QoS Standard Priority Queue Statistics 25-15

CHAPTER 26 **Configuring Application Layer Protocol Inspection** 26-1
Inspection Engine Overview 26-2
When to Use Application Protocol Inspection 26-2
Inspection Limitations 26-2
Default Inspection Policy 26-3
Configuring Application Inspection 26-5
CTIQBE Inspection 26-10
CTIQBE Inspection Overview 26-10

Limitations and Restrictions 26-10
Verifying and Monitoring CTIQBE Inspection 26-10
DCERPC Inspection 26-12
DCERPC Overview 26-12
Configuring a DCERPC Inspection Policy Map for Additional Inspection Control 26-13
DNS Inspection 26-14
How DNS Application Inspection Works 26-14
How DNS Rewrite Works 26-15
Configuring DNS Rewrite 26-16
Using the Static Command for DNS Rewrite 26-16
Using the Alias Command for DNS Rewrite 26-17
Configuring DNS Rewrite with Two NAT Zones 26-17
DNS Rewrite with Three NAT Zones 26-18
Configuring DNS Rewrite with Three NAT Zones 26-20
Verifying and Monitoring DNS Inspection 26-21
Configuring a DNS Inspection Policy Map for Additional Inspection Control 26-21
ESMTP Inspection 26-24
Configuring an ESMTP Inspection Policy Map for Additional Inspection Control 26-25
FTP Inspection 26-27
FTP Inspection Overview 26-28
Using the **strict** Option 26-28
Configuring an FTP Inspection Policy Map for Additional Inspection Control 26-29
Verifying and Monitoring FTP Inspection 26-32
GTP Inspection 26-33
GTP Inspection Overview 26-33
Configuring a GTP Inspection Policy Map for Additional Inspection Control 26-34
Verifying and Monitoring GTP Inspection 26-38
H.323 Inspection 26-39
H.323 Inspection Overview 26-39
How H.323 Works 26-40
Limitations and Restrictions 26-41
Configuring an H.323 Inspection Policy Map for Additional Inspection Control 26-41
Configuring H.323 and H.225 Timeout Values 26-44
Verifying and Monitoring H.323 Inspection 26-44
Monitoring H.225 Sessions 26-44
Monitoring H.245 Sessions 26-45
Monitoring H.323 RAS Sessions 26-45
HTTP Inspection 26-46
HTTP Inspection Overview 26-46

Configuring an HTTP Inspection Policy Map for Additional Inspection Control 26-46
Instant Messaging Inspection 26-50
IM Inspection Overview 26-50
Configuring an Instant Messaging Inspection Policy Map for Additional Inspection Control 26-50
ICMP Inspection 26-53
ICMP Error Inspection 26-54
ILS Inspection 26-54
MGCP Inspection 26-55
MGCP Inspection Overview 26-55
Configuring an MGCP Inspection Policy Map for Additional Inspection Control 26-57
Configuring MGCP Timeout Values 26-58
Verifying and Monitoring MGCP Inspection 26-58
MMP Inspection 26-59
Configuring MMP Inspection for a TLS Proxy 26-60
NetBIOS Inspection 26-61
Configuring a NetBIOS Inspection Policy Map for Additional Inspection Control 26-61
PPTP Inspection 26-63
RADIUS Accounting Inspection 26-63
Configuring a RADIUS Inspection Policy Map for Additional Inspection Control 26-64
RSH Inspection 26-64
RTSP Inspection 26-65
RTSP Inspection Overview 26-65
Using RealPlayer 26-65
Restrictions and Limitations 26-66
Configuring an RTSP Inspection Policy Map for Additional Inspection Control 26-66
Configuring a SIP Inspection Policy Map for Additional Inspection Control 26-66
SIP Inspection 26-68
SIP Inspection Overview 26-69
SIP Instant Messaging 26-69
Configuring a SIP Inspection Policy Map for Additional Inspection Control 26-70
Configuring SIP Timeout Values 26-74
Verifying and Monitoring SIP Inspection 26-74
Skinny (SCCP) Inspection 26-75
SCCP Inspection Overview 26-75
Supporting Cisco IP Phones 26-75
Restrictions and Limitations 26-76
Verifying and Monitoring SCCP Inspection 26-76
Configuring a Skinny (SCCP) Inspection Policy Map for Additional Inspection Control 26-77

SMTP and Extended SMTP Inspection 26-78
SNMP Inspection 26-80
SQL*Net Inspection 26-80
Sun RPC Inspection 26-81
Sun RPC Inspection Overview 26-81
Managing Sun RPC Services 26-81
Verifying and Monitoring Sun RPC Inspection 26-82
TFTP Inspection 26-83
XDMCP Inspection 26-84

CHAPTER 27 **Configuring Cisco Unified Communications Proxy Features** 27-1
Overview of the Adaptive Security Appliance in Cisco Unified Communications 27-1
TLS Proxy Applications in Cisco Unified Communications 27-3
Licensing for Cisco Unified Communications Proxy Features 27-4
Phone Proxy 27-5
About the Phone Proxy 27-5
Phone Proxy Limitations and Restrictions 27-7
Phone Proxy Configuration 27-8
Configuration Prerequisites 27-9
Requirements to Support the 7960 and 7940 IP Phones 27-11
Addressing Requirements for IP Phones on Multiple Interfaces 27-11
Supported Cisco UCM and IP Phones for the Phone Proxy 27-12
End-User Phone Provisioning 27-13
Configuring the Phone Proxy in a Non-secure Cisco UCM Cluster 27-13
Importing Certificates from the Cisco UCM 27-17
Configuring the Phone Proxy in a Mixed-mode Cisco UCM Cluster 27-19
Phone Proxy Configuration for Cisco IP Communicator 27-24
Configuring Linksys Routers for UDP Port Forwarding 27-24
About Rate Limiting TFTP Requests 27-25
About ICMP Traffic Destined for the Media Termination Address 27-26
Troubleshooting the Phone Proxy 27-26
Debugging Information from the Security Appliance 27-26
Debugging Information from IP Phones 27-30
IP Phone Registration Failure 27-31
Media Termination Address Errors 27-40
Audio Problems with IP Phones 27-40
Saving SAST Keys 27-41
TLS Proxy for Encrypted Voice Inspection 27-42
Overview 27-43

Configuring TLS Proxy 27-43
Debugging TLS Proxy 27-47
CTL Client 27-50
Cisco Unified Mobility and MMP Inspection Engine 27-52
Mobility Proxy Overview 27-52
Mobility Proxy Deployment Scenarios 27-53
Establishing Trust Relationships for Cisco UMA Deployments 27-56
Configuring the Security Appliance for Cisco Unified Mobility 27-57
Debugging for Cisco Unified Mobility 27-58
Cisco Unified Presence 27-59
Architecture for Cisco Unified Presence 27-59
Establishing a Trust Relationship in the Presence Federation 27-61
About the Security Certificate Exchange Between Cisco UP and the Security Appliance 27-62
Configuring the Presence Federation Proxy for Cisco Unified Presence 27-62
Debugging the Security Appliance for Cisco Unified Presence 27-64
Sample Configurations for Cisco Unified Communications Proxy Features 27-65
Phone Proxy Sample Configurations 27-65
Example 1: Nonsecure Cisco UCM cluster, Cisco UCM and TFTP Server on Publisher 27-65
Example 2: Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Publisher 27-66
Example 3: Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Different Servers 27-68
Example 4: Mixed-mode Cisco UCM cluster, Primary Cisco UCM, Secondary and TFTP Server on Different Servers 27-69
Example 5: LSC Provisioning in Mixed-mode Cisco UCM cluster; Cisco UCM and TFTP Server on Publisher 27-71
Example 6: VLAN Transversal 27-73
Cisco Unified Mobility Sample Configurations 27-75
Example 1: Cisco UMC/Cisco UMA Architecture – Security Appliance as Firewall with TLS Proxy and MMP Inspection 27-75
Example 2: Cisco UMC/Cisco UMA Architecture – Security Appliance as TLS Proxy Only 27-76
Cisco Unified Presence Sample Configuration 27-78

CHAPTER 28 **Configuring ARP Inspection and Bridging Parameters for Transparent Mode** 28-1

Configuring ARP Inspection 28-1
ARP Inspection Overview 28-1
Adding a Static ARP Entry 28-2
Enabling ARP Inspection 28-2
Customizing the MAC Address Table 28-3
MAC Address Table Overview 28-3
Adding a Static MAC Address 28-3

Setting the MAC Address Timeout 28-4
Disabling MAC Address Learning 28-4
Viewing the MAC Address Table 28-4

PART 3 Configuring VPN

CHAPTER 29 Configuring IPsec and ISAKMP 29-1
Tunneling Overview 29-1
IPsec Overview 29-2
Configuring ISAKMP 29-2
ISAKMP Overview 29-2
Configuring ISAKMP Policies 29-5
Enabling ISAKMP on the Outside Interface 29-6
Disabling ISAKMP in Aggressive Mode 29-6
Determining an ID Method for ISAKMP Peers 29-6
Enabling IPsec over NAT-T 29-7
Using NAT-T 29-7
Enabling IPsec over TCP 29-8
Waiting for Active Sessions to Terminate Before Rebooting 29-8
Alerting Peers Before Disconnecting 29-9
Configuring Certificate Group Matching 29-9
Creating a Certificate Group Matching Rule and Policy 29-9
Using the Tunnel-group-map default-group Command 29-11
Configuring IPsec 29-11
Understanding IPsec Tunnels 29-11
Understanding Transform Sets 29-12
Defining Crypto Maps 29-12
Applying Crypto Maps to Interfaces 29-19
Using Interface Access Lists 29-19
Changing IPsec SA Lifetimes 29-21
Creating a Basic IPsec Configuration 29-22
Using Dynamic Crypto Maps 29-23
Providing Site-to-Site Redundancy 29-26
Viewing an IPsec Configuration 29-26
Clearing Security Associations 29-26
Clearing Crypto Map Configurations 29-27
Supporting the Nokia VPN Client 29-27

CHAPTER 30 **Configuring L2TP over IPSec** 30-1

L2TP Overview 30-1

IPSec Transport and Tunnel Modes 30-2

Configuring L2TP over IPSec Connections 30-3

Tunnel Group Switching 30-5

Apple iPhone and MAC OS X Compatibility 30-6

Viewing L2TP over IPSec Connection Information 30-6

Using L2TP Debug Commands 30-8

Enabling IPSec Debug 30-9

Getting Additional Information 30-9

CHAPTER 31 **Setting General IPSec VPN Parameters** 31-1

Configuring VPNs in Single, Routed Mode 31-1

Configuring IPSec to Bypass ACLs 31-1

Permitting Intra-Interface Traffic 31-2

NAT Considerations for Intra-Interface Traffic 31-3

Setting Maximum Active IPSec VPN Sessions 31-3

Using Client Update to Ensure Acceptable Client Revision Levels 31-4

Understanding Load Balancing 31-6

Implementing Load Balancing 31-6

Prerequisites 31-7

Eligible Platforms 31-7

Eligible Clients 31-7

VPN Load-Balancing Cluster Configurations 31-7

Some Typical Mixed Cluster Scenarios 31-8

Scenario 1: Mixed Cluster with No WebVPN Connections 31-8

Scenario 2: Mixed Cluster Handling WebVPN Connections 31-8

Configuring Load Balancing 31-9

Configuring the Public and Private Interfaces for Load Balancing 31-9

Configuring the Load Balancing Cluster Attributes 31-10

Enabling Redirection Using a Fully-qualified Domain Name 31-11

Viewing Load Balancing 31-12

Configuring VPN Session Limits 31-13

CHAPTER 32 **Configuring Connection Profiles, Group Policies, and Users** 32-1

Overview of Connection Profiles, Group Policies, and Users 32-1

Connection Profiles 32-2

General Connection Profile Connection Parameters 32-3

IPSec Tunnel-Group Connection Parameters 32-4
Connection Profile Connection Parameters for Clientless SSL VPN Sessions 32-5
Configuring Connection Profiles 32-6
Maximum Connection Profiles 32-6
Default IPSec Remote Access Connection Profile Configuration 32-7
Configuring IPSec Tunnel-Group General Attributes 32-7
Configuring IPSec Remote-Access Connection Profiles 32-8
Specifying a Name and Type for the IPSec Remote Access Connection Profile 32-8
Configuring IPSec Remote-Access Connection Profile General Attributes 32-8
Enabling IPv6 VPN Access 32-12
Configuring IPSec Remote-Access Connection Profile IPSec Attributes 32-13
Configuring IPSec Remote-Access Connection Profile PPP Attributes 32-15
Configuring LAN-to-LAN Connection Profiles 32-16
Default LAN-to-LAN Connection Profile Configuration 32-16
Specifying a Name and Type for a LAN-to-LAN Connection Profile 32-16
Configuring LAN-to-LAN Connection Profile General Attributes 32-17
Configuring LAN-to-LAN IPSec Attributes 32-17
Configuring Connection Profiles for Clientless SSL VPN Sessions 32-19
Specifying a Connection Profile Name and Type for Clientless SSL VPN Sessions 32-19
Configuring General Tunnel-Group Attributes for Clientless SSL VPN Sessions 32-20
Configuring Tunnel-Group Attributes for Clientless SSL VPN Sessions 32-23
Customizing Login Windows for Users of Clientless SSL VPN sessions 32-27
Configuring Microsoft Active Directory Settings for Password Management 32-27
Using Active Directory to Force the User to Change Password at Next Logon 32-28
Using Active Directory to Specify Maximum Password Age 32-30
Using Active Directory to Override an Account Disabled AAA Indicator 32-31
Using Active Directory to Enforce Minimum Password Length 32-32
Using Active Directory to Enforce Password Complexity 32-33
Configuring the Connection Profile for RADIUS/SDI Message Support for the AnyConnect Client 32-34
AnyConnect Client and RADIUS/SDI Server Interaction 32-34
Configuring the Security Appliance to Support RADIUS/SDI Messages 32-35
Group Policies 32-36
Default Group Policy 32-37
Configuring Group Policies 32-38
Configuring an External Group Policy 32-38
Configuring an Internal Group Policy 32-39
Configuring Group Policy Attributes 32-40
Configuring WINS and DNS Servers 32-40
Configuring VPN-Specific Attributes 32-41

Configuring Security Attributes 32-44
Configuring the Banner Message 32-46
Configuring IPSec-UDP Attributes 32-46
Configuring Split-Tunneling Attributes 32-47
Configuring Domain Attributes for Tunneling 32-48
Configuring Attributes for VPN Hardware Clients 32-50
Configuring Backup Server Attributes 32-53
Configuring Microsoft Internet Explorer Client Parameters 32-54
Configuring Network Admission Control Parameters 32-56
Configuring Address Pools 32-60
Configuring Firewall Policies 32-60
Configuring Client Access Rules 32-63
Configuring Group-Policy Attributes for Clientless SSL VPN Sessions 32-65
Configuring User Attributes 32-75
Viewing the Username Configuration 32-76
Configuring Attributes for Specific Users 32-76
Setting a User Password and Privilege Level 32-76
Configuring User Attributes 32-77
Configuring VPN User Attributes 32-77
Configuring Clientless SSL VPN Access for Specific Users 32-81

CHAPTER 33 **Configuring IP Addresses for VPNs** 33-1
Configuring an IP Address Assignment Method 33-1
Configuring Local IP Address Pools 33-2
Configuring AAA Addressing 33-2
Configuring DHCP Addressing 33-3

CHAPTER 34 **Configuring Remote Access IPSec VPNs** 34-1
Summary of the Configuration 34-1
Configuring Interfaces 34-2
Configuring ISAKMP Policy and Enabling ISAKMP on the Outside Interface 34-3
Configuring an Address Pool 34-4
Adding a User 34-4
Creating a Transform Set 34-4
Defining a Tunnel Group 34-5
Creating a Dynamic Crypto Map 34-6
Creating a Crypto Map Entry to Use the Dynamic Crypto Map 34-7

CHAPTER 35 **Configuring Network Admission Control** 35-1

Overview 35-1

Uses, Requirements, and Limitations 35-2

Viewing the NAC Policies on the Security Appliance 35-2

Adding, Accessing, or Removing a NAC Policy 35-4

Configuring a NAC Policy 35-4

Specifying the Access Control Server Group 35-5

Setting the Query-for-Posture-Changes Timer 35-5

Setting the Revalidation Timer 35-6

Configuring the Default ACL for NAC 35-6

Configuring Exemptions from NAC 35-7

Assigning a NAC Policy to a Group Policy 35-8

Changing Global NAC Framework Settings 35-8

Changing Clientless Authentication Settings 35-8

Enabling and Disabling Clientless Authentication 35-8

Changing the Login Credentials Used for Clientless Authentication 35-9

Changing NAC Framework Session Attributes 35-10

CHAPTER 36 **Configuring Easy VPN Services on the ASA 5505** 36-1

Specifying the Client/Server Role of the Cisco ASA 5505 36-1

Specifying the Primary and Secondary Servers 36-2

Specifying the Mode 36-3

NEM with Multiple Interfaces 36-3

Configuring Automatic Xauth Authentication 36-4

Configuring IPSec Over TCP 36-4

Comparing Tunneling Options 36-5

Specifying the Tunnel Group or Trustpoint 36-6

Specifying the Tunnel Group 36-7

Specifying the Trustpoint 36-7

Configuring Split Tunneling 36-8

Configuring Device Pass-Through 36-8

Configuring Remote Management 36-9

Guidelines for Configuring the Easy VPN Server 36-9

Group Policy and User Attributes Pushed to the Client 36-10

Authentication Options 36-12

CHAPTER 37 **Configuring the PPPoE Client** 37-1

PPPoE Client Overview 37-1

Configuring the PPPoE Client Username and Password 37-2

Enabling PPPoE 37-3

Using PPPoE with a Fixed IP Address 37-3

Monitoring and Debugging the PPPoE Client 37-4

Clearing the Configuration 37-5

Using Related Commands 37-5

CHAPTER 38 **Configuring LAN-to-LAN IPsec VPNs** 38-1

Summary of the Configuration 38-1

Configuring Interfaces 38-2

Configuring ISAKMP Policy and Enabling ISAKMP on the Outside Interface 38-2

Creating a Transform Set 38-4

Configuring an ACL 38-4

Defining a Tunnel Group 38-5

Creating a Crypto Map and Applying It To an Interface 38-6

Applying Crypto Maps to Interfaces 38-7

CHAPTER 39 **Configuring Clientless SSL VPN** 39-1

Getting Started 39-1

Observing Clientless SSL VPN Security Precautions 39-2

Understanding Features Not Supported in Clientless SSL VPN 39-3

Using SSL to Access the Central Site 39-3

Using HTTPS for Clientless SSL VPN Sessions 39-3

Configuring Clientless SSL VPN and ASDM Ports 39-4

Configuring Support for Proxy Servers 39-4

Configuring SSL/TLS Encryption Protocols 39-6

Authenticating with Digital Certificates 39-6

Enabling Cookies on Browsers for Clientless SSL VPN 39-7

Managing Passwords 39-7

Using Single Sign-on with Clientless SSL VPN 39-8

Configuring SSO with HTTP Basic or NTLM Authentication 39-9

Configuring SSO Authentication Using SiteMinder 39-10

Configuring SSO Authentication Using SAML Browser Post Profile 39-12

Configuring SSO with the HTTP Form Protocol 39-15

Authenticating with Digital Certificates 39-21

Creating and Applying Clientless SSL VPN Resources 39-21
Assigning Users to Group Policies 39-21
Using the Security Appliance Authentication Server 39-21
Using a RADIUS Server 39-21
Configuring Connection Profile Attributes for Clientless SSL VPN 39-22
Configuring Group Policy and User Attributes for Clientless SSL VPN 39-22
Configuring Browser Access to Client-Server Plug-ins 39-24
Introduction to Browser Plug-Ins 39-24
Plug-in Requirements and Restrictions 39-25
Preparing the Security Appliance for a Plug-in 39-25
Installing Plug-ins Redistributed By Cisco 39-26
Providing Access to Third-Party Plug-ins 39-28
Example: Providing Access to a Citrix Java Presentation Server 39-28
Viewing the Plug-ins Installed on the Security Appliance 39-29
Configuring Application Access 39-30
Configuring Smart Tunnel Access 39-30
About Smart Tunnels 39-30
Why Smart Tunnels? 39-31
Smart Tunnel Requirements, Restrictions, and Limitations 39-31
Adding Applications to Be Eligible for Smart Tunnel Access 39-32
Assigning a Smart Tunnel List 39-35
Configuring Smart Tunnel Auto Sign-on 39-36
Automating Smart Tunnel Access 39-37
Enabling and Disabling Smart Tunnel Access 39-38
Configuring Port Forwarding 39-38
About Port Forwarding 39-39
Why Port Forwarding? 39-39
Port Forwarding Requirements and Restrictions 39-39
Configuring DNS for Port Forwarding 39-40
Adding Applications to Be Eligible for Port Forwarding 39-41
Assigning a Port Forwarding List 39-42
Automating Port Forwarding 39-43
Enabling and Disabling Port Forwarding 39-43
Application Access User Notes 39-44
Using Application Access on Vista 39-44
Closing Application Access to Prevent hosts File Errors 39-44
Recovering from hosts File Errors When Using Application Access 39-44
Configuring File Access 39-47
Adding Support for File Access 39-48

Ensuring Clock Accuracy for SharePoint Access 39-49
Using Clientless SSL VPN with PDAs 39-50
Using E-Mail over Clientless SSL VPN 39-50
Configuring E-mail Proxies 39-50
E-mail Proxy Certificate Authentication 39-51
Configuring Web E-mail: MS Outlook Web Access 39-51
Optimizing Clientless SSL VPN Performance 39-52
Configuring Caching 39-52
Configuring Content Transformation 39-52
Configuring a Certificate for Signing Rewritten Java Content 39-53
Disabling Content Rewrite 39-53
Using Proxy Bypass 39-53
Configuring Application Profile Customization Framework 39-54
APCF Syntax 39-54
APCF Example 39-56
Clientless SSL VPN End User Setup 39-56
Defining the End User Interface 39-56
Viewing the Clientless SSL VPN Home Page 39-57
Viewing the Clientless SSL VPN Application Access Panel 39-57
Viewing the Floating Toolbar 39-58
Customizing Clientless SSL VPN Pages 39-59
How Customization Works 39-59
Exporting a Customization Template 39-60
Editing the Customization Template 39-60
Importing a Customization Object 39-66
Applying Customizations to Connection Profiles, Group Policies and Users 39-66
Login Screen Advanced Customization 39-67
Customizing Help 39-71
Customizing a Help File Provided By Cisco 39-72
Creating Help Files for Languages Not Provided by Cisco 39-73
Importing a Help File to Flash Memory 39-73
Exporting a Previously Imported Help File from Flash Memory 39-74
Requiring Usernames and Passwords 39-74
Communicating Security Tips 39-75
Configuring Remote Systems to Use Clientless SSL VPN Features 39-75
Translating the Language of User Messages 39-79
Understanding Language Translation 39-80
Creating Translation Tables 39-81
Referencing the Language in a Customization Object 39-82

Changing a Group Policy or User Attributes to Use the Customization Object 39-84
Capturing Data 39-84
Creating a Capture File 39-84
Using a Browser to Display Capture Data 39-85

CHAPTER 40 **Configuring AnyConnect VPN Client Connections** 40-1
Installing the AnyConnect SSL VPN Client 40-2
Remote PC System Requirements 40-2
Installing the AnyConnect Client 40-2
Enabling AnyConnect Client Connections 40-3
Enabling Permanent Client Installation 40-5
Configuring DTLS 40-5
Prompting Remote Users 40-6
Enabling AnyConnect Client Profile Downloads 40-6
Enabling Additional AnyConnect Client Features 40-8
Enabling Start Before Logon 40-9
Translating Languages for AnyConnect User Messages 40-9
Understanding Language Translation 40-10
Creating Translation Tables 40-10
Configuring Advanced SSL VPN Features 40-12
Enabling Rekey 40-12
Enabling and Adjusting Dead Peer Detection 40-12
Enabling Keepalive 40-13
Using Compression 40-14
Adjusting MTU Size 40-14
Viewing SSL VPN Sessions 40-15
Logging Off SVC Sessions 40-15
Updating SSL VPN Client Images 40-16

CHAPTER 41 **Configuring Certificates** 41-1
Public Key Cryptography 41-1
About Public Key Cryptography 41-1
Certificate Scalability 41-2
About Key Pairs 41-2
About Trustpoints 41-3
About Revocation Checking 41-3
About CRLs 41-3
About OCSP 41-4

Supported CA Servers 41-5
Certificate Configuration 41-5
Preparing for Certificates 41-5
Configuring Key Pairs 41-6
Generating Key Pairs 41-6
Removing Key Pairs 41-7
Configuring Trustpoints 41-7
Obtaining Certificates 41-9
Obtaining Certificates with SCEP 41-9
Obtaining Certificates Manually 41-11
Using Extended Keys for Certificates 41-13
Configuring CRLs for a Trustpoint 41-13
Exporting and Importing Trustpoints 41-15
Exporting a Trustpoint Configuration 41-15
Importing a Trustpoint Configuration 41-16
Configuring CA Certificate Map Rules 41-16
The Local CA 41-17
Configuring the Local CA Server 41-18
The Default Local CA Server 41-19
Customizing the Local CA Server 41-20
Certificate Characteristics 41-21
Defining Storage for Local CA Files 41-23
Default Flash Memory Data Storage 41-23
Setting up External Local CA File Storage 41-24
CRL Storage 41-24
CRL Downloading 41-25
Enrolling Local CA Users 41-25
Setting Up Enrollment Parameters 41-27
Enrollment Requirements 41-27
Starting and Stopping the Local CA Server 41-28
Enabling the Local CA Server 41-28
Debugging the Local CA Server 41-29
Disabling the Local CA Server 41-29
Managing the Local CA User Database 41-29
Adding and Enrolling Users 41-30
Renewing Users 41-31
Revoking Certificates and Removing or Restoring Users 41-31
Revocation Checking 41-32
Displaying Local CA Server Information 41-32
Display Local CA Configuration 41-33

Display Certificate Database 41-33
Display the Local CA Certificate 41-34
Display the CRL 41-34
Display the User Database 41-34
Local CA Server Maintenance and Backup Procedures 41-35
Maintaining the Local CA User Database 41-35
Maintaining the Local CA Certificate Database 41-36
Local CA Certificate Rollover 41-36
Archiving the Local CA Server Certificate and Keypair 41-36
Deleting the Local CA Server 41-37

PART 4 System Administration

CHAPTER 42 Managing System Access 42-1
Allowing Telnet Access 42-1
Allowing SSH Access 42-2
Configuring SSH Access 42-2
Using an SSH Client 42-3
Allowing HTTPS Access for ASDM 42-3
Enabling HTTPS Access 42-4
Accessing ASDM from Your PC 42-4
Managing the Security Appliance on a Different Interface from the VPN Tunnel Termination Interface 42-5
Configuring AAA for System Administrators 42-5
Configuring Authentication for CLI and ASDM Access 42-5
Configuring Authentication To Access Privileged EXEC Mode (the enable Command) 42-6
Configuring Authentication for the enable Command 42-6
Authenticating Users Using the Login Command 42-7
Limiting User CLI and ASDM Access with Management Authorization 42-7
Configuring Command Authorization 42-8
Command Authorization Overview 42-9
Configuring Local Command Authorization 42-11
Configuring TACACS+ Command Authorization 42-14
Configuring Command Accounting 42-18
Viewing the Current Logged-In User 42-18
Recovering from a Lockout 42-19
Configuring a Login Banner 42-20

CHAPTER 43 **Managing Software and Configurations** 43-1

Viewing Files in Flash Memory 43-1

Retrieving Files from Flash Memory 43-2

Removing Files from Flash Memory 43-2

Downloading Software or Configuration Files to Flash Memory 43-2

Downloading a File to a Specific Location 43-3

Downloading a File to the Startup or Running Configuration 43-3

Configuring the Application Image and ASDM Image to Boot 43-4

Configuring the File to Boot as the Startup Configuration 43-5

Performing Zero Downtime Upgrades for Failover Pairs 43-5

Upgrading an Active/Standby Failover Configuration 43-6

Upgrading and Active/Active Failover Configuration 43-7

Backing Up Configuration Files 43-8

Backing up the Single Mode Configuration or Multiple Mode System Configuration 43-8

Backing Up a Context Configuration in Flash Memory 43-8

Backing Up a Context Configuration within a Context 43-9

Copying the Configuration from the Terminal Display 43-9

Backing Up Additional Files Using the Export and Import Commands 43-9

Using a Script to Back Up and Restore Files 43-10

Prerequisites 43-10

Running the Script 43-10

Sample Script 43-11

Configuring Auto Update Support 43-19

Configuring Communication with an Auto Update Server 43-20

Configuring Client Updates as an Auto Update Server 43-21

Viewing Auto Update Status 43-22

CHAPTER 44 **Monitoring the Security Appliance** 44-1

Using SNMP 44-1

SNMP Overview 44-1

Enabling SNMP 44-4

Configuring and Managing Logs 44-5

Logging Overview 44-6

Logging in Multiple Context Mode 44-6

Analyzing Syslogs 44-6

Enabling and Disabling Logging 44-7

Enabling Logging to All Configured Output Destinations 44-7

Disabling Logging to All Configured Output Destinations 44-7

Viewing the Log Configuration 44-7
Configuring Log Output Destinations 44-8
Sending System Log Messages to a Syslog Server 44-8
Sending System Log Messages to the Console Port 44-10
Sending System Log Messages to an E-mail Address 44-10
Sending System Log Messages to ASDM 44-11
Sending System Log Messages to a Telnet or SSH Session 44-13
Sending System Log Messages to the Log Buffer 44-14
Filtering System Log Messages 44-16
Message Filtering Overview 44-17
Filtering System Log Messages by Class 44-17
Filtering System Log Messages with Custom Message Lists 44-18
Customizing the Log Configuration 44-19
Configuring the Logging Queue 44-20
Including the Date and Time in System Log Messages 44-20
Including the Device ID in System Log Messages 44-20
Generating System Log Messages in EMBLEM Format 44-21
Disabling a System Log Message 44-22
Changing the Severity Level of a System Log Message 44-22
Limiting the Rate of System Log Message Generation 44-23
Changing the Amount of Internal Flash Memory Available for Logs 44-23
Understanding System Log Messages 44-24
System Log Message Format 44-24
Severity Levels 44-25

CHAPTER 45 **Troubleshooting the Security Appliance** 45-1
Testing Your Configuration 45-1
Enabling ICMP Debug Messages and System Log Messages 45-1
Pinging Security Appliance Interfaces 45-2
Pinging Through the Security Appliance 45-4
Disabling the Test Configuration 45-5
Traceroute 45-6
Packet Tracer 45-6
Reloading the Security Appliance 45-6
Performing Password Recovery 45-6
Recovering Passwords for the ASA 5500 Series Adaptive Security Appliance 45-7
Recovering Passwords for the PIX 500 Series Security Appliance 45-8
Disabling Password Recovery 45-9
Resetting the Password on the SSM Hardware Module 45-10

Using the ROM Monitor to Load a Software Image 45-10
Erasing the Flash File System 45-12
Other Troubleshooting Tools 45-12
Viewing Debug Messages 45-12
Capturing Packets 45-12
Viewing the Crash Dump 45-13
Common Problems 45-13

PART 5 Reference

APPENDIX A Sample Configurations A-1
Example 1: Multiple Mode Firewall With Outside Access A-1
System Configuration for Example 1 A-3
Admin Context Configuration for Example 1 A-4
Customer A Context Configuration for Example 1 A-4
Customer B Context Configuration for Example 1 A-5
Customer C Context Configuration for Example 1 A-5
Example 2: Single Mode Firewall Using Same Security Level A-6
Example 3: Shared Resources for Multiple Contexts A-8
System Configuration for Example 3 A-9
Admin Context Configuration for Example 3 A-10
Department 1 Context Configuration for Example 3 A-11
Department 2 Context Configuration for Example 3 A-12
Example 4: Multiple Mode, Transparent Firewall with Outside Access A-13
System Configuration for Example 4 A-14
Admin Context Configuration for Example 4 A-15
Customer A Context Configuration for Example 4 A-16
Customer B Context Configuration for Example 4 A-16
Customer C Context Configuration for Example 4 A-17
Example 5: Single Mode, Transparent Firewall with NAT A-18
Example 6: IPv6 Configuration A-19
Example 7: Dual ISP Support Using Static Route Tracking A-20
Example 8: Multicast Routing A-21
For PIM Sparse Mode A-22
For PIM bidir Mode A-23
Example 9: LAN-Based Active/Standby Failover (Routed Mode) A-24
Primary Unit Configuration for Example 9 A-24
Secondary Unit Configuration for Example 9 A-25

Example 10: LAN-Based Active/Active Failover (Routed Mode) A-25
Primary Unit Configuration for Example 10 A-26
Primary System Configuration for Example 10 A-26
Primary admin Context Configuration for Example 10 A-27
Primary ctx1 Context Configuration for Example 10 A-28
Secondary Unit Configuration for Example 10 A-28
Example 11: LAN-Based Active/Standby Failover (Transparent Mode) A-28
Primary Unit Configuration for Example 11 A-29
Secondary Unit Configuration for Example 11 A-30
Example 12: LAN-Based Active/Active Failover (Transparent Mode) A-30
Primary Unit Configuration for Example 12 A-31
Primary System Configuration for Example 12 A-31
Primary admin Context Configuration for Example 12 A-32
Primary ctx1 Context Configuration for Example 12 A-33
Secondary Unit Configuration for Example 12 A-33
Example 13: Cable-Based Active/Standby Failover (Routed Mode) A-34
Example 14: Cable-Based Active/Standby Failover (Transparent Mode) A-35
Example 15: ASA 5505 Base License A-36
Example 16: ASA 5505 Security Plus License with Failover and Dual-ISP Backup A-38
Primary Unit Configuration for Example 16 A-38
Secondary Unit Configuration for Example 16 A-40
Example 17: AIP SSM in Multiple Context Mode A-40
System Configuration for Example 17 A-41
Context 1 Configuration for Example 17 A-42
Context 2 Configuration for Example 17 A-42
Context 3 Configuration for Example 17 A-43

APPENDIX B **Using the Command-Line Interface** B-1
Firewall Mode and Security Context Mode B-1
Command Modes and Prompts B-2
Syntax Formatting B-3
Abbreviating Commands B-3
Command-Line Editing B-3
Command Completion B-4
Command Help B-4
Filtering show Command Output B-4
Command Output Paging B-6
Adding Comments B-7

Text Configuration Files B-7
How Commands Correspond with Lines in the Text File B-7
Command-Specific Configuration Mode Commands B-7
Automatic Text Entries B-8
Line Order B-8
Commands Not Included in the Text Configuration B-8
Passwords B-8
Multiple Security Context Files B-8
Supported Character Sets B-9

APPENDIX C **Addresses, Protocols, and Ports** C-1
IPv4 Addresses and Subnet Masks C-1
Classes C-1
Private Networks C-2
Subnet Masks C-2
Determining the Subnet Mask C-3
Determining the Address to Use with the Subnet Mask C-3
IPv6 Addresses C-5
IPv6 Address Format C-5
IPv6 Address Types C-6
Unicast Addresses C-6
Multicast Address C-8
Anycast Address C-9
Required Addresses C-10
IPv6 Address Prefixes C-10
Protocols and Applications C-11
TCP and UDP Ports C-11
Local Ports and Protocols C-14
ICMP Types C-15

APPENDIX D **Configuring an External Server for Authorization and Authentication** D-1
Understanding Policy Enforcement of Permissions and Attributes D-2
Configuring an External LDAP Server D-3
Organizing the Security Appliance for LDAP Operations D-3
Searching the Hierarchy D-4
Binding the Security Appliance to the LDAP Server D-5
Login DN Example for Active Directory D-5
Defining the Security Appliance LDAP Configuration D-5
Supported Cisco Attributes for LDAP Authorization D-6

Cisco-AV-Pair Attribute Syntax D-12
Active Directory/LDAP VPN Remote Access Authorization Use Cases D-14
User-Based Attributes Policy Enforcement D-15
Placing LDAP users in a specific Group-Policy D-17
Enforcing Static IP Address Assignment for AnyConnect Tunnels D-19
Enforcing Dial-in Allow or Deny Access D-22
Enforcing Logon Hours and Time-of-Day Rules D-25
Configuring an External RADIUS Server D-27
Reviewing the RADIUS Configuration Procedure D-27
Security Appliance RADIUS Authorization Attributes D-27
Configuring an External TACACS+ Server D-35

APPENDIX E **Configuring the Security Appliance for Use with MARS** E-1
Taskflow for Configuring MARS to Monitor Security Appliances E-1
Enabling Administrative Access to MARS on the Security Appliance E-2
Adding a Security Appliance to Monitor E-3
Adding Security Contexts E-4
Adding Discovered Contexts E-4
Editing Discovered Contexts E-5
Setting the Logging Severity Level for System Log Messages E-5
System Log Messages That Are Processed by MARS E-5
Configuring Specific Features E-7

GLOSSARY

INDEX



About This Guide

This preface introduces the *Cisco Security Appliance Command Line Configuration Guide*, and includes the following sections:

- Document Objectives, page xli
- Audience, page xli
- Related Documentation, page xlii
- Document Conventions, page xlii
- Obtaining Documentation, Obtaining Support, and Security Guidelines, page xlii

Document Objectives

The purpose of this guide is to help you configure the security appliance using the command-line interface. This guide does not cover every feature, but describes only the most common configuration scenarios.

You can also configure and monitor the security appliance by using ASDM, a GUI application. ASDM includes configuration wizards to guide you through some common configuration scenarios, and online Help for less common scenarios. For more information, see:
http://www.cisco.com/en/US/products/ps6121/tsd_products_support_series_home.html

For software Versions 8.0(4) and below, this guide applies to the Cisco PIX 500 series security appliances (PIX 515E, PIX 525, and PIX 535). The PIX security appliance is not supported in Version 8.0(5) and above.

For all software versions, this guide applies to the Cisco ASA 5500 series security appliances (ASA 5505, ASA 5510, ASA 5520, ASA 5540, and ASA 5550). The ASA 5580 is not supported in Version 8.0.

Throughout this guide, the term “security appliance” applies generically to all supported models, unless specified otherwise.

Note The PIX 501, PIX 506E, and PIX 520 security appliances are not supported.

Audience

This guide is for network managers who perform any of the following tasks:

- Manage network security
- Install and configure firewalls/security appliances
- Configure VPNs
- Configure intrusion detection software

Related Documentation

For more information, refer to *Navigating the Cisco ASA 5500 Series Documentation*:

http://www.cisco.com/en/US/docs/security/asa/roadmap/asaroadmap.html

Document Conventions

Command descriptions use these conventions:

- Braces ({ }) indicate a required choice.
- Square brackets ([ ]) indicate optional elements.
- Vertical bars (|) separate alternative, mutually exclusive elements.
- **Boldface** indicates commands and keywords that are entered literally as shown.
- *Italics* indicate arguments for which you supply values.

Examples use these conventions:

- Examples depict screen displays and the command line in `screen` font.
- Information you need to enter in examples is shown in **`boldface screen`** font.
- Variables for which you must supply a value are shown in *`italic screen`* font.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Obtaining Documentation, Obtaining Support, and Security Guidelines

For information on obtaining documentation, obtaining support, providing documentation feedback, security guidelines, and also recommended aliases and general Cisco documents, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html



PART 1

Getting Started and General Information



CHAPTER 1

Introduction to the Security Appliance

The security appliance combines advanced stateful firewall and VPN concentrator functionality in one device, and for some models, an integrated intrusion prevention module called the AIP SSM or an integrated content security and control module called the CSC SSM. The security appliance includes many advanced features, such as multiple security contexts (similar to virtualized firewalls), transparent (Layer 2) firewall or routed (Layer 3) firewall operation, advanced inspection engines, IPSec and clientless SSL support, and many more features. See the "Supported Feature Licenses Per Model" section on page 3-1 for a list of supported platforms and features. For a list of new features, see the *Cisco ASA 5500 Series Release Notes* or the *Cisco PIX Security Appliance Release Notes*.

This chapter includes the following sections:

- Supported Platform Models, page 1-1
- SSM and SSC Support Per Model, page 1-2
- VPN Specifications, page 1-3
- New Features, page 1-3
- Firewall Functional Overview, page 1-14
- VPN Functional Overview, page 1-18
- Security Context Overview, page 1-19

Supported Platform Models

Software Version 8.0 is supported on the following platform models:

- ASA 5505
- ASA 5510
- ASA 5520
- ASA 5540
- ASA 5550
- PIX 515/515E
- PIX 525
- PIX 535

Note The Cisco PIX 501 and PIX 506E security appliances are not supported in any version; all other PIX models are supported in Version 8.0(4) and earlier only.

The ASA 5580 is not supported in Version 8.0.

For information about licenses and features supported on each platform, see Chapter 3, "Managing Feature Licenses."

SSM and SSC Support Per Model

Table 1-1 shows the SSMs supported by each platform:

Table 1-1 SSM Support

Platform	SSM Models
ASA 5505	No support
ASA 5510	AIP SSM 10 AIP SSM 20 CSC SSM 10 CSC SSM 20 4GE SSM
ASA 5520	AIP SSM 10 AIP SSM 20 CSC SSM 10 CSC SSM 20 4GE SSM
ASA 5540	AIP SSM 10 AIP SSM 20 CSC SSM 10[1] CSC SSM 20[1] 4GE SSM
ASA 5550	No support (the 4GE SSM is built-in and not user-removable)

1. The CSC SSM licenses support up to 1000 users while the Cisco ASA 5540 Series appliance can support significantly more users. If you deploy CSC SSM with an ASA 5540 adaptive security appliance, be sure to configure the security appliance to send the CSC SSM only the traffic that should be scanned.

VPN Specifications

See the *Cisco ASA 5500 Series VPN Compatibility Reference* at http://cisco.cisco.com/en/US/docs/security/asa/compatibility/asa-vpn-compatibility.html.

New Features

This section lists the features added for each maintenance release, and includes the following topics:

- New Features in Version 8.0(5), page 1-3
- New Features in Version 8.0(4), page 1-4
- New Features in Version 8.0(3), page 1-8
- New Features in Version 8.0(2), page 1-9

New Features in Version 8.0(5)

Table 1-2 lists the new features for Version 8.0(5).

Note Version 8.0(5) is not supported on the PIX security appliance.

Table 1-2 New Features for ASA Version 8.0(5)

Feature	Description
Remote Access Features	
Scalable Solutions for Waiting-to-Resume VPN Sessions	An administrator can now keep track of the number of users in the active state and can look at the statistics. The sessions that have been inactive for the longest time are marked as idle (and are automatically logged off) so that license capacity is not reached and new users can log in The following ASDM screen was modified: Monitoring > VPN > VPN Statistics > Sessions.
Application Inspection Features	
Enabling Call Set up Between H.323 Endpoints	You can enable call setup between H.323 endpoints when the Gatekeeper is inside the network. The security appliance includes options to open pinholes for calls based on the RegistrationRequest/RegistrationConfirm (RRQ/RCF) messages. Because these RRQ/RCF messages are sent to and from the Gatekeeper, the calling endpoint's IP address is unknown and the security appliance opens a pinhole through source IP address/port 0/0. By default, this option is disabled. The following commands were introduced: **ras-enhancement enable**, **show running-configuration ras-enhancement**, **clear configure ras-enhancement**. The following ASDM screen was modified: Configuration > Firewall > Objects > Inspect Maps > H.323 > Details > State Checking.
Interface Features	

Table 1-2 New Features for ASA Version 8.0(5) (continued)

Feature	Description
In multiple context mode, auto-generated MAC addresses now use a user-configurable prefix, and other enhancements	The MAC address format was changed to use a prefix, to use a fixed starting value (A2), and to use a different scheme for the primary and secondary unit MAC addresses in a failover pair. The MAC addresess are also now persistent accross reloads. The command parser now checks if auto-generation is enabled; if you want to also manually assign a MAC address, you cannot start the manual MAC address with A2. The following command was modified: **mac-address auto prefix** *prefix*. The following ASDM screen was modified: Configuration > Context Management > Security Contexts.
High Availablility Features	
No notifications when interfaces are brought up or brought down during a switchover event	To distinguish between link up/down transitions during normal operation from link up/down transitions during failover, no link up/link down traps are sent during a failover. Also, no related syslog messages are sent.
Routing Features	
DHCP RFC compatibility (rfc3011, rfc3527) to resolve routing issues	This enhancement introduces security appliance support for DHCP RFCs 3011 (The IPv4 Subnet Selection Option) and 3527 (Link Selection Sub-option for the Relay Agent Information Option). For each DHCP server that is configured using the **dhcp-server** command, you can now configure the security appliance to send the **subnet-selection** option, and the **link-selection** option or neither. The following ASDM screen was modified: Remote Access VPN > Network Access > IPsec connection profiles > Add/Edit.

New Features in Version 8.0(4)

Table 1-3 lists the new features for Version 8.0(4).

Table 1-3 New Features for ASA and PIX Version 8.0(4)

Feature	Description
Unified Communications Features[1]	
Phone Proxy	Phone Proxy functionality is supported. ASA Phone Proxy provides similar features to those of the Metreos Cisco Unified Phone Proxy with additional support for SIP inspection and enhanced security. The ASA Phone Proxy has the following key features: • Secures remote IP phones by forcing the phones to encrypt signaling and media • Performs certificate-based authentication with remote IP phones • Terminates TLS signaling from IP phones and initiates TCP and TLS to Cisco Unified Mobility Advantage servers • Terminates SRTP and initiates RTP/SRTP to the called party In ASDM, see Configuration > Firewall > Advanced > Encrypted Traffic Inspection > Phone Proxy.

Table 1-3 New Features for ASA and PIX Version 8.0(4) (continued)

Feature	Description
Mobility Proxy	Secure connectivity (mobility proxy) between Cisco Unified Mobility Advantage clients and servers is supported. Cisco Unified Mobility Advantage solutions include the Cisco Unified Mobile Communicator, an easy-to-use software application for mobile handsets that extends enterprise communications applications and services to mobile phones and smart phones and the Cisco Unified Mobility Advantage server. The mobility solution streamlines the communication experience, enabling real-time collaboration across the enterprise. The ASA in this solution delivers inspection for the MMP (formerly called OLWP) protocol, the proprietary protocol between Cisco Unified Mobile Communicator and Cisco Unified Mobility Advantage. The ASA also acts as a TLS proxy, terminating and reoriginating the TLS signaling between the Cisco Unified Mobile Communicator and Cisco Unified Mobility Advantage. In ASDM, see Configuration > Firewall > Advanced > Encrypted Traffic Inspection > TLS Proxy.
Presence Federation Proxy	Secure connectivity (presence federation proxy) between Cisco Unified Presence servers and Cisco/Microsoft Presence servers is supported. With the Presence solution, businesses can securely connect their Cisco Unified Presence clients back to their enterprise networks, or share Presence information between Presence servers in different enterprises. The ASA delivers functionality to enable Presence for Internet and intra-enterprise communications. An SSL-enabled Cisco Unified Presence client can establish an SSL connection to the Presence Server. The ASA enables SSL connectivity between server to server communication including third-party Presence servers communicating with Cisco Unified Presence servers. Enterprises share Presence information, and can use IM applications. The ASA inspects SIP messages between the servers. In ASDM, see Configuration > Firewall > Service Policy Rules > Add/Edit Service Policy Rule > Rule Actions > Protocol Inspection or Configuration > Firewall > Advanced > Encrypted Traffic Inspection > TLS Proxy > Add > Client Configuration.
Remote Access Features	
Auto Sign-On with Smart Tunnels for IE[1]	This feature lets you enable the replacement of logon credentials for WININET connections. Most Microsoft applications use WININET, including Internet Explorer. Mozilla Firefox does not, so it is not supported by this feature. It also supports HTTP-based authentication, therefore form-based authentication does not work with this feature. Credentials are statically associated to destination hosts, not services, so if initial credentials are wrong, they cannot be dynamically corrected during runtime. Also, because of the association with destinations hosts, providing support for an auto sign-on enabled host may not be desirable if you want to deny access to some of the services on that host. To configure a group auto sign-on for smart tunnels, you create a global list of auto sign-on sites, then assign the list to group policies or user names. This feature is not supported with Dynamic Access Policy. In ASDM, see Firewall > Advanced > ACL Manager.
Entrust Certificate Provisioning[1]	ASDM includes a link to the Entrust website to apply for temporary (test) or discounted permanent SSL identity certificates for your ASA. In ASDM, see Configuration > Remote Access VPN > Certificate Management > Identity Certificates. Click **Enroll ASA SSL VPN head-end with Entrust**.

Table 1-3 New Features for ASA and PIX Version 8.0(4) (continued)

Feature	Description
Extended Time for User Reauthentication on IKE Rekey	You can configure the security appliance to give remote users more time to enter their credentials on a Phase 1 SA rekey. Previously, when reauthenticate-on-rekey was configured for IKE tunnels and a phase 1 rekey occurred, the security appliance prompted the user to authenticate and only gave the user approximately 2 minutes to enter their credentials. If the user did not enter their credentials in that 2 minute window, the tunnel would be terminated. With this new feature enabled, users now have more time to enter credentials before the tunnel drops. The total amount of time is the difference between the new Phase 1 SA being established, when the rekey actually takes place, and the old Phase 1 SA expiring. With default Phase 1 rekey times set, the difference is roughly 3 hours, or about 15% of the rekey interval. In ASDM, see Configuration > Device Management > Certificate Management > Identity Certificates.
Persistent IPsec Tunneled Flows	With the persistent IPsec tunneled flows feature enabled, the security appliance preserves and resumes stateful (TCP) tunneled flows after the tunnel drops, then recovers. All other flows are dropped when the tunnel drops and must reestablish when a new tunnel comes up. Preserving the TCP flows allows some older or sensitive applications to keep working through a short-lived tunnel drop. This feature supports IPsec LAN-to-LAN tunnels and Network Extension Mode tunnels from a Hardware Client. It does not support IPsec or AnyConnect/SSL VPN remote access tunnels. See the **[no] sysopt connection preserve-vpn-flows** command. This option is disabled by default. In ASDM, see Configuration > Remote Access VPN > Network (Client) Access > Advanced > IPsec > System Options. Check the **Preserve stateful VPN flows when the tunnel drops for Network Extension Mode (NEM)** checkbox to enable persistent IPsec tunneled flows.
Show Active Directory Groups	The CLI command **show ad-groups** was added to list the active directory groups. ASDM Dynamic Access Policy uses this command to present the administrator with a list of MS AD groups that can be used to define the VPN policy. In ASDM, see Configuration > Remote Access VPN > Clientless SSL VPN Access > Dynamic Access Policies > Add/Edit DAP > Add/Edit AAA Attribute.
Smart Tunnel over Mac OS[1]	Smart tunnels now support Mac OS. In ASDM, see Configuration > Remote Access VPN > Clientless SSL VPN Access > Portal > Smart Tunnels.
Firewall Features	
QoS Traffic Shaping	If you have a device that transmits packets at a high speed, such as the security appliance with Fast Ethernet, and it is connected to a low speed device such as a cable modem, then the cable modem is a bottleneck at which packets are frequently dropped. To manage networks with differing line speeds, you can configure the security appliance to transmit packets at a fixed slower rate. See the **shape** command. See also the **crypto ipsec security-association replay** command, which lets you configure the IPSec anti-replay window size. One side-effect of priority queueing is packet re-ordering. For IPSec packets, out-of-order packets that are not within the anti-replay window generate warning syslog messages. These warnings become false alarms in the case of priority queueing. This new command avoids possible false alarms. In ASDM, see Configuration > Firewall > Security Policy > Service Policy Rules > Add/Edit Service Policy Rule > Rule Actions > QoS. Note that the only traffic class supported for traffic shaping is class-default, which matches all traffic.

Table 1-3 New Features for ASA and PIX Version 8.0(4) (continued)

Feature	Description
TCP Normalization Enhancements	You can now configure TCP normalization actions for certain packet types. Previously, the default actions for these kinds of packets was to drop the packet. Now you can set the TCP normalizer to allow the packets. • TCP invalid ACK check (the **invalid-ack** command) • TCP packet sequence past window check (the **seq-past-window** command) • TCP SYN-ACK with data check (the **synack-data** command) You can also set the TCP out-of-order packet buffer timeout (the **queue** command **timeout** keyword). Previously, the timeout was 4 seconds. You can now set the timeout to another value. The default action for packets that exceed MSS has changed from drop to allow (the **exceed-mss** command). The following non-configurable actions have changed from drop to clear for these packet types: • Bad option length in TCP • TCP Window scale on non-SYN • Bad TCP window scale value • Bad TCP SACK ALLOW option In ASDM, see Configuration > Firewall > Objects > TCP Maps.
TCP Intercept statistics	You can enable collection for TCP Intercept statistics using the **threat-detection statistics tcp-intercept** command, and view them using the **show threat-detection statistics** command. In ASDM 6.1(5) and later, see Configuration > Firewall > Threat Detection. This command was not supported in ASDM 6.1(3).
Threat detection shun timeout	You can now configure the shun timeout for threat detection using the **threat-detection scanning-threat shun duration** command. In ASDM 6.1(5) and later, see Configuration > Firewall > Threat Detection. This command was not supported in ASDM 6.1(3).
Timeout for SIP Provisional Media	You can now configure the timeout for SIP provisional media using the **timeout sip-provisional-media** command. In ASDM, see Configuration > Firewall > Advanced > Global Timeouts.
Platform Features	
Native VLAN support for the ASA 5505	You can now include the native VLAN in an ASA 5505 trunk port using the **switchport trunk native vlan** command. In ASDM, see Configuration > Device Setup > Interfaces > Switch Ports > Edit dialog.
SNMP support for unnamed interfaces	Previously, SNMP only provided information about interfaces that were configured using the **nameif** command. For example, SNMP only sent traps and performed walks on the IF MIB and IP MIB for interfaces that were named. Because the ASA 5505 has both unnamed switch ports and named VLAN interfaces, SNMP was enhanced to show information about all physical interfaces and logical interfaces; a **nameif** command is no longer required to display the interfaces using SNMP. These changes affect all models, and not just the ASA 5505.

1. This feature is not supported on the PIX security appliance.

New Features in Version 8.0(3)

Table 1-4 lists the new features for Version 8.0(3).

Table 1-4 New Features for ASA and PIX Version 8.0(3)

Feature	Description
AnyConnect RSA SoftID API Integration	Provides support for AnyConnect VPN clients to communicate directly with RSA SoftID for obtaining user token codes. It also provides the ability to specify SoftID message support for a connection profile (tunnel group), and the ability to configure SDI messages on the security appliance that match SDI messages received through a RADIUS proxy. This feature ensures the prompts displayed to the remote client user are appropriate for the action required during authentication and the AnyConnect client responds successfully to authentication challenges.
IP Address Reuse Delay	Delays the reuse of an IP address after it has been returned to the IP address pool. Increasing the delay prevents problems the security appliance may experience when an IP address is returned to the pool and reassigned quickly. In ASDM, see Configure > Remote Access VPN > Network (Client) Access > Address Assignment > Assignment Policy.
WAAS Inspection	Added support for Wide Area Application Services (WAAS) inspection. WAAS gives branch and remote offices LAN-like access to WAN and MAN services. See the **inspect waas** command. In ASDM, see Configuration > Firewall > Service Policy Rules > Add/Edit Service Policy Rule > Rule Actions > Protocol Inspection.
DNS Guard Enhancement	Added an option to enable or disable DNS guard. When enabled, this feature allows only one DNS response back from a DNS request. In ASDM, see Configuration > Firewall > Objects > Inspect maps > DNS.
Fully Qualified Domain Name Support Enhancement	Added option in the **redirect-fqdn** command to send either the fully qualified domain name (FQDN) or the IP address to the client in a VPN load balancing cluster. In ASDM, see Configuration > Device Management >High Availability > VPN Load Balancing or Configuration > Remote Access VPN >Load Balancing.
Clientless SSL VPN Caching Static Content Enhancement	Added a new command to allow clientless SSL VPN users to cache the static content, **cache-static-content** enable. In ASDM, see Configuration > Remote Access VPN > Clientless SSL VPN Access > Advanced > Content Cache.
DHCP Client Enhancements	Added two new items for the DHCP client. The first option configures DHCP Option 61 to send either the MAC or the string "cisco-<MAC>-<interface>-<hostname>", where < > represents the corresponding values as the client identifier. The second option either sets or clears the broadcast flag for DHCP discover when the DHCP request has the broadcast flag enabled. In ASDM, see Configuration > Device Management > DHCP > DHCP Server; then click on Advanced button.
ASDM Banner	When you start ASDM, new banner text appears in a dialog box with the option to continue or disconnect. See the **banner asdm** command. In ASDM, see Configuration > Properties > Device Administration > Banner.

Table 1-4 New Features for ASA and PIX Version 8.0(3) (continued)

Feature	Description
ESMTP Enhancement	Addedan option for Extended Simple Mail Transfer Protocol (ESMTP) inspection to work over Transport Layer Security (TLS). In ASDM, see Configuration > Firewall > Objects > Inspect Map > ESMTP.
Smart Card Removal Enhancement	Added option in the VPN group policy to specify whether tunnels stay connected or not when the Smart Card is removed. Previously, the tunnels were always disconnected. See the **smartcard-removal-disconnect command**. In ASDM, see Configuration > Remote Access VPN > Network (Client) Access > Group Policies > Add/Edit Internal/External Group Policies > More Options.

New Features in Version 8.0(2)

Table 1-5 lists the new features for Version 8.0(2).

Note There was no ASA or PIX 8.0(1) release.

Table 1-5 New Features for ASA and PIX Version 8.0(2)

ASA Feature Type	Feature	Description
General Features		
Routing	EIGRP routing	The security appliance supports EIGRP or EIGRP stub routing.
High Availability	Remote command execution in Failover pairs	You can execute commands on the peer unit in a failover pair without having to connect directly to the peer. This works for both Active/Standby and Active/Active failover.
	CSM configuration rollback support	Adds support for the Cisco Security Manager configuration rollback feature in failover configurations.
	Failover pair Auto Update support	You can use an Auto Update server to update the platform image and configuration in failover pairs.
	Stateful Failover for SIP signaling	SIP media and signaling connections are replicated to the standby unit.
	Redundant interfaces	A logical redundant interface pairs an active and a standby physical interface. When the active interface fails, the standby interface becomes active and starts passing traffic. You can configure a redundant interface to increase the security appliance reliability. This feature is separate from device-level failover, but you can configure redundant interfaces as well as failover if desired. You can configure up to eight redundant interface pairs.
SSMs	Password reset	You can reset the password on the SSM hardware module.

Table 1-5 New Features for ASA and PIX Version 8.0(2) (continued)

ASA Feature Type	Feature	Description
VPN Features[1]		
Authentication Enhancements	Combined certificate and username/password login	An administrator requires a username and password in addition to a certificate for login to SSL VPN connections.
	Internal domain username/password	Provides a password for access to internal resources for users who log in with credentials other than a domain username and password, for example, with a one-time password. This is a password in addition to the one a user enters when logging in.
	Generic LDAP support	This includes OpenLDAP and Novell LDAP. Expands LDAP support available for authentication and authorization.
	Onscreen keyboard	The security appliance includes an onscreen keyboard option for the login page and subsequent authentication requests for internal resources. This provides additional protection against software-based keystroke loggers by requiring a user to use a mouse to click characters in an onscreen keyboard for authentication, rather than entering the characters on a physical keyboard.
	SAML SSO verified with RSA Access Manager	The security appliance supports Security Assertion Markup Language (SAML) protocol for Single Sign On (SSO) with RSA Access Manager (Cleartrust and Federated Identity Manager).
	NTLMv2	Version 8.0(2) adds support for NTLMv2 authentication for Windows-based clients.
Certificates	Local certificate authority	Provides a certificate authority on the security appliance for use with SSL VPN connections, both browser- and client-based.
	OCSP CRL	Provides OCSP revocation checking for SSL VPN.
Cisco Secure Desktop	Host Scan	As a condition for the completion of a Cisco AnyConnect or clientless SSL VPN connection, the remote computer scans for a greatly expanded collection of antivirus and antispyware applications, firewalls, operating systems, and associated updates. It also scans for any registry entries, filenames, and process names that you specify. It sends the scan results to the security appliance. The security appliance uses both the user login credentials and the computer scan results to assign a Dynamic Access Policy (DAP). With an Advanced Endpoint Assessment License, you can enhance Host Scan by configuring an attempt to update noncompliant computers to meet version requirements. Cisco can provide timely updates to the list of applications and versions that Host Scan supports in a package that is separate from Cisco Secure Desktop.
	Simplified prelogin assessment and periodic checks	Cisco Secure Desktop now simplifies the configuration of prelogin and periodic checks to perform on remote Microsoft Windows computers. Cisco Secure Desktop lets you add, modify, remove, and place conditions on endpoint checking criteria using a simplified, graphical view of the checks. As you use this graphical view to configure sequences of checks, link them to branches, deny logins, and assign endpoint profiles, Cisco Secure Desktop Manager records the changes to an XML file. You can configure the security appliance to use returned results in combination with many other types of data, such as the connection type and multiple group settings, to generate and apply a DAP to the session.

Table 1-5 New Features for ASA and PIX Version 8.0(2) (continued)

ASA Feature Type	Feature	Description
Access Policies	Dynamic access policies (DAP)	VPN gateways operate in dynamic environments. Multiple variables can affect each VPN connection, for example, intranet configurations that frequently change, the various roles each user may inhabit within an organization, and logins from remote access sites with different configurations and levels of security. The task of authorizing users is much more complicated in a VPN environment than it is in a network with a static configuration. Dynamic Access Policies (DAP) on the security appliance let you configure authorization that addresses these many variables. You create a dynamic access policy by setting a collection of access control attributes that you associate with a specific user tunnel or session. These attributes address issues of multiple group membership and endpoint security. That is, the security appliance grants access to a particular user for a particular session based on the policies you define. It generates a DAP at the time the user connects by selecting and/or aggregating attributes from one or more DAP records. It selects these DAP records based on the endpoint security information of the remote device and the AAA authorization information for the authenticated user. It then applies the DAP record to the user tunnel or session.
	Administrator differentiation	Lets you differentiate regular remote access users and administrative users under the same database, either RADIUS or LDAP. You can create and restrict access to the console via various methods (TELNET and SSH, for example) to administrators only. It is based on the IETF RADIUS service-type attribute.
Platform Enhancements	VLAN support for remote access VPN connections	Provides support for mapping (tagging) of client traffic at the group or user level. This feature is compatible with clientless as well as IPsec and SSL tunnel-based connections.
	VPN load balancing for the ASA 5510	Extends load balancing support to ASA 5510 adaptive security appliances that have a Security Plus license.
	Crypto conditional debug	Lets users debug an IPsec tunnel on the basis of predefined crypto conditions such as the peer IP address, connection-ID of a crypto engine, and security parameter index (SPI). By limiting debug messages to specific IPSec operations and reducing the amount of debug output, you can better troubleshoot the security appliance with a large number of tunnels.

Table 1-5 New Features for ASA and PIX Version 8.0(2) (continued)

ASA Feature Type	Feature	Description
Browser-based SSL VPN Features	Enhanced portal design	Version 8.0(2) includes an enhanced end user interface that is more cleanly organized and visually appealing.
	Customization	Supports administrator-defined customization of all user-visible content.
	Support for FTP	You can provide file access via FTP in additional to CIFS (Windows-based).
	Plugin applets	Version 8.0(2) adds a framework for supporting TCP-based applications without requiring a pre-installed client application. Java applets let users access these applications from the browser-enabled SSL VPN portal. Initial support is for TELNET, SSH, RDP, and VNC.
	Smart tunnels	A smart tunnel is a connection between an application and a remote site, using a browser-based SSL VPN session with the security appliance as the pathway. Version 8.0(2) lets you identify the applications to which you want to grant smart tunnel access, and lets you specify the path to the application and the SHA-1 hash of its checksum to check before granting it access. Lotus SameTime and Microsoft Outlook Express are examples of applications to which you might want to grant smart tunnel access. The remote host originating the smart tunnel connection must be running Microsoft Windows Vista, Windows XP, or Windows 2000, and the browser must be enabled with Java, Microsoft ActiveX, or both.
	RSS newsfeed	Administrators can populate the clientless portal with RSS newsfeed information, which lets company news or other information display on a user screen.
Browser-based SSL VPN Features (continued)	Personal bookmark support	Users can define their own bookmarks. These bookmarks are stored on a file server.
	Transformation enhancements	Adds support for several complex forms of web content over clientless connections, including Adobe flash and Java WebStart.
	IPv6	Allows access to IPv6 resources over a public IPv4 connection.
	Web folders	Lets browser-based SSL VPN users connecting from Windows operating systems browse shared file systems and perform the following operations: view folders, view folder and file properties, create, move, copy, copy from the local host to the remote host, copy from the remote host to the local host, and delete. Internet Explorer indicates when a web folder is accessible. Accessing this folder launches another window, providing a view of the shared folder, on which users can perform web folder functions, assuming the properties of the folders and documents permit them.
	Microsoft Sharepoint enhancement	Extends Web Access support for Microsoft Sharepoint, integrating Microsoft Office applications available on the machine with the browser to view, change, and save documents shared on a server. Version 8.0(2) supports Windows Sharepoint Services 2.0 in Windows Server 2003.
HTTP Proxy	PAC support	Lets you specify the URL of a proxy autoconfiguration file (PAC) to download to the browser. Once downloaded, the PAC file uses a JavaScript function to identify a proxy for each URL.
HTTPS Proxy	Proxy exclusion list	Lets you configure a list of URLs to exclude from the HTTP requests the security appliance can send to an external proxy server.

Table 1-5 New Features for ASA and PIX Version 8.0(2) (continued)

ASA Feature Type	Feature	Description
NAC	SSL VPN tunnel support	The security appliance provides NAC posture validation of endpoints that establish AnyConnect VPN client sessions.
	Support for audit services	You can configure the security appliance to pass the IP address of the client to an optional audit server if the client does not respond to a posture validation request. The audit server uses the host IP address to challenge the host directly to assess its health. For example, it might challenge the host to determine whether its virus checking software is active and up-to-date. After the audit server completes its interaction with the remote host, it passes a token to the posture validation server, indicating the health of the remote host. If the token indicates the remote host is healthy, the posture validation server sends a network access policy to the security appliance for application to the traffic on the tunnel.
Firewall Features		
Application Inspection	Modular policy framework inspect class map	Traffic can match one of multiple match commands in an inspect class map; formerly, traffic had to match all match commands in a class map to match the class map.
	AIC for encrypted streams and AIC Arch changes	Provides HTTP inspection into TLS, which allows AIC/MPF inspection in WebVPN HTTP and HTTPS streams.
	TLS Proxy for SCCP and SIP[2]	Enables inspection of encrypted traffic. Implementations include SSL encrypted VoIP signaling, namely Skinny and SIP, interacting with the Cisco CallManager.
	SIP enhancements for CCM	Improves interoperability with CCM 5.0 and 6.x with respect to signaling pinholes.
	Full RTSP PAT support	Provides TCP fragment reassembly support, a scalable parsing routine on RTSP, and security enhancements that protect RTSP traffic.
Access Lists	Enhanced service object group	Lets you configure a service object group that contains a mix of TCP services, UDP services, ICMP-type services, and any protocol. It removes the need for a specific ICMP-type object group and protocol object group. The enhanced service object group also specifies both source and destination services. The access list CLI now supports this behavior.
	Ability to rename access list	Lets you rename an access list.
	Live access list hit counts	Includes the hit count for ACEs from multiple access lists. The hit count value represents how many times traffic hits a particular access rule.
Attack Prevention	Set connection limits for management traffic to the adaptive security appliance	For a Layer 3/4 management class map, you can specify the **set connection** command.
	Threat detection	You can enable basic threat detection and scanning threat detection to monitor attacks such as DoS attacks and scanning attacks. For scanning attacks, you can automatically shun attacking hosts. You can also enable scan threat statistics to monitor both valid and invalid traffic for hosts, ports, protocols, and access lists.

Table 1-5 New Features for ASA and PIX Version 8.0(2) (continued)

ASA Feature Type	Feature	Description
NAT	Transparent firewall NAT support	You can configure NAT for a transparent firewall.
IPS	Virtual IPS sensors with the AIP SSM	The AIP SSM running IPS software Version 6.0 and above can run multiple virtual sensors, which means you can configure multiple security policies on the AIP SSM. You can assign each context or single mode adaptive security appliance to one or more virtual sensors, or you can assign multiple security contexts to the same virtual sensor. See the IPS documentation for more information about virtual sensors, including the maximum number of sensors supported.
Logging	Secure logging	You can enable secure connections to the syslog server using SSL or TLS with TCP, and encrypted system log message content. Not supported on the PIX series adaptive security appliance.
IPv6	IPv6 support for SIP	The SIP inspection engine supports IPv6 addresses. IPv6 addresses can be used in URLs, in the Via header field, and SDP fields.

1. Clientless SSL VPN features are not supported on the PIX security appliance.
2. TLS proxy is not supported on the PIX security appliance.

Firewall Functional Overview

Firewalls protect inside networks from unauthorized access by users on an outside network. A firewall can also protect inside networks from each other, for example, by keeping a human resources network separate from a user network. If you have network resources that need to be available to an outside user, such as a web or FTP server, you can place these resources on a separate network behind the firewall, called a *demilitarized zone* (DMZ). The firewall allows limited access to the DMZ, but because the DMZ only includes the public servers, an attack there only affects the servers and does not affect the other inside networks. You can also control when inside users access outside networks (for example, access to the Internet), by allowing only certain addresses out, by requiring authentication or authorization, or by coordinating with an external URL filtering server.

When discussing networks connected to a firewall, the *outside* network is in front of the firewall, the *inside* network is protected and behind the firewall, and a *DMZ*, while behind the firewall, allows limited access to outside users. Because the security appliance lets you configure many interfaces with varied security policies, including many inside interfaces, many DMZs, and even many outside interfaces if desired, these terms are used in a general sense only.

This section includes the following topics:

- Security Policy Overview, page 1-15
- Firewall Mode Overview, page 1-17
- Stateful Inspection Overview, page 1-17

Security Policy Overview

A security policy determines which traffic is allowed to pass through the firewall to access another network. By default, the security appliance allows traffic to flow freely from an inside network (higher security level) to an outside network (lower security level). You can apply actions to traffic to customize the security policy. This section includes the following topics:

- Permitting or Denying Traffic with Access Lists, page 1-15
- Applying NAT, page 1-15
- Protecting from IP Fragments, page 1-15
- Using AAA for Through Traffic, page 1-15
- Applying HTTP, HTTPS, or FTP Filtering, page 1-16
- Applying Application Inspection, page 1-16
- Sending Traffic to the Advanced Inspection and Prevention Security Services Module, page 1-16
- Sending Traffic to the Content Security and Control Security Services Module, page 1-16
- Applying QoS Policies, page 1-16
- Applying Connection Limits and TCP Normalization, page 1-16

Permitting or Denying Traffic with Access Lists

You can apply an access list to limit traffic from inside to outside, or allow traffic from outside to inside. For transparent firewall mode, you can also apply an EtherType access list to allow non-IP traffic.

Applying NAT

Some of the benefits of NAT include the following:

- You can use private addresses on your inside networks. Private addresses are not routable on the Internet.
- NAT hides the local addresses from other networks, so attackers cannot learn the real address of a host.
- NAT can resolve IP routing problems by supporting overlapping IP addresses.

Protecting from IP Fragments

The security appliance provides IP fragment protection. This feature performs full reassembly of all ICMP error messages and virtual reassembly of the remaining IP fragments that are routed through the security appliance. Fragments that fail the security check are dropped and logged. Virtual reassembly cannot be disabled.

Using AAA for Through Traffic

You can require authentication and/or authorization for certain types of traffic, for example, for HTTP. The security appliance also sends accounting information to a RADIUS or TACACS+ server.

Applying HTTP, HTTPS, or FTP Filtering

Although you can use access lists to prevent outbound access to specific websites or FTP servers, configuring and managing web usage this way is not practical because of the size and dynamic nature of the Internet. We recommend that you use the security appliance in conjunction with a separate server running one of the following Internet filtering products:

- Websense Enterprise
- Secure Computing SmartFilter

Applying Application Inspection

Inspection engines are required for services that embed IP addressing information in the user data packet or that open secondary channels on dynamically assigned ports. These protocols require the security appliance to do a deep packet inspection.

Sending Traffic to the Advanced Inspection and Prevention Security Services Module

If your model supports the AIP SSM for intrusion prevention, then you can send traffic to the AIP SSM for inspection. The AIP SSM is an intrusion prevention services module that monitors and performs real-time analysis of network traffic by looking for anomalies and misuse based on an extensive, embedded signature library. When the system detects unauthorized activity, it can terminate the specific connection, permanently block the attacking host, log the incident, and send an alert to the device manager. Other legitimate connections continue to operate independently without interruption. For more information, see *Configuring the Cisco Intrusion Prevention System Sensor Using the Command Line Interface*.

Sending Traffic to the Content Security and Control Security Services Module

If your model supports it, the CSC SSM provides protection against viruses, spyware, spam, and other unwanted traffic. It accomplishes this by scanning the FTP, HTTP, POP3, and SMTP traffic that you configure the adaptive security appliance to send to it.

Applying QoS Policies

Some network traffic, such as voice and streaming video, cannot tolerate long latency times. QoS is a network feature that lets you give priority to these types of traffic. QoS refers to the capability of a network to provide better service to selected network traffic.

Applying Connection Limits and TCP Normalization

You can limit TCP and UDP connections and embryonic connections. Limiting the number of connections and embryonic connections protects you from a DoS attack. The security appliance uses the embryonic limit to trigger TCP Intercept, which protects inside systems from a DoS attack perpetrated by flooding an interface with TCP SYN packets. An embryonic connection is a connection request that has not finished the necessary handshake between source and destination.

TCP normalization is a feature consisting of advanced TCP connection settings designed to drop packets that do not appear normal.

Enabling Threat Detection

You can configure scanning threat detection and basic threat detection, and also how to use statistics to analyze threats.

Basic threat detection detects activity that might be related to an attack, such as a DoS attack, and automatically sends a system log message.

A typical scanning attack consists of a host that tests the accessibility of every IP address in a subnet (by scanning through many hosts in the subnet or sweeping through many ports in a host or subnet). The scanning threat detection feature determines when a host is performing a scan. Unlike IPS scan detection that is based on traffic signatures, the security appliance scanning threat detection feature maintains an extensive database that contains host statistics that can be analyzed for scanning activity.

The host database tracks suspicious activity such as connections with no return activity, access of closed service ports, vulnerable TCP behaviors such as non-random IPID, and many more behaviors.

You can configure the security appliance to send system log messages about an attacker or you can automatically shun the host.

Firewall Mode Overview

The security appliance runs in two different firewall modes:

- Routed
- Transparent

In routed mode, the security appliance is considered to be a router hop in the network.

In transparent mode, the security appliance acts like a “bump in the wire,” or a “stealth firewall,” and is not considered a router hop. The security appliance connects to the same network on its inside and outside interfaces.

You might use a transparent firewall to simplify your network configuration. Transparent mode is also useful if you want the firewall to be invisible to attackers. You can also use a transparent firewall for traffic that would otherwise be blocked in routed mode. For example, a transparent firewall can allow multicast streams using an EtherType access list.

Stateful Inspection Overview

All traffic that goes through the security appliance is inspected using the Adaptive Security Algorithm and either allowed through or dropped. A simple packet filter can check for the correct source address, destination address, and ports, but it does not check that the packet sequence or flags are correct. A filter also checks every packet against the filter, which can be a slow process.

A stateful firewall like the security appliance, however, takes into consideration the state of a packet:

- Is this a new connection?

  If it is a new connection, the security appliance has to check the packet against access lists and perform other tasks to determine if the packet is allowed or denied. To perform this check, the first packet of the session goes through the “session management path,” and depending on the type of traffic, it might also pass through the “control plane path.”

  The session management path is responsible for the following tasks:

  – Performing the access list checks

  - Performing route lookups
  - Allocating NAT translations (xlates)
  - Establishing sessions in the “fast path”

Note The session management path and the fast path make up the “accelerated security path.”

Some packets that require Layer 7 inspection (the packet payload must be inspected or altered) are passed on to the control plane path. Layer 7 inspection engines are required for protocols that have two or more channels: a data channel, which uses well-known port numbers, and a control channel, which uses different port numbers for each session. These protocols include FTP, H.323, and SNMP.

- Is this an established connection?

  If the connection is already established, the security appliance does not need to re-check packets; most matching packets can go through the fast path in both directions. The fast path is responsible for the following tasks:

  - IP checksum verification
  - Session lookup
  - TCP sequence number check
  - NAT translations based on existing sessions
  - Layer 3 and Layer 4 header adjustments

  For UDP or other connectionless protocols, the security appliance creates connection state information so that it can also use the fast path.

  Data packets for protocols that require Layer 7 inspection can also go through the fast path.

  Some established session packets must continue to go through the session management path or the control plane path. Packets that go through the session management path include HTTP packets that require inspection or content filtering. Packets that go through the control plane path include the control packets for protocols that require Layer 7 inspection.

VPN Functional Overview

A VPN is a secure connection across a TCP/IP network (such as the Internet) that appears as a private connection. This secure connection is called a tunnel. The security appliance uses tunneling protocols to negotiate security parameters, create and manage tunnels, encapsulate packets, transmit or receive them through the tunnel, and unencapsulate them. The security appliance functions as a bidirectional tunnel endpoint: it can receive plain packets, encapsulate them, and send them to the other end of the tunnel where they are unencapsulated and sent to their final destination. It can also receive encapsulated packets, unencapsulate them, and send them to their final destination. The security appliance invokes various standard protocols to accomplish these functions.

The security appliance performs the following functions:

- Establishes tunnels
- Negotiates tunnel parameters
- Authenticates users
- Assigns user addresses
- Encrypts and decrypts data

- Manages security keys
- Manages data transfer across the tunnel
- Manages data transfer inbound and outbound as a tunnel endpoint or router

The security appliance invokes various standard protocols to accomplish these functions.

Security Context Overview

You can partition a single security appliance into multiple virtual devices, known as security contexts. Each context is an independent device, with its own security policy, interfaces, and administrators. Multiple contexts are similar to having multiple standalone devices. Many features are supported in multiple context mode, including routing tables, firewall features, IPS, and management. Some features are not supported, including VPN and dynamic routing protocols.

In multiple context mode, the security appliance includes a configuration for each context that identifies the security policy, interfaces, and almost all the options you can configure on a standalone device. The system administrator adds and manages contexts by configuring them in the system configuration, which, like a single mode configuration, is the startup configuration. The system configuration identifies basic settings for the security appliance. The system configuration does not include any network interfaces or network settings for itself; rather, when the system needs to access network resources (such as downloading the contexts from the server), it uses one of the contexts that is designated as the admin context.

The admin context is just like any other context, except that when a user logs into the admin context, then that user has system administrator rights and can access the system and all other contexts.

Note You can run all your contexts in routed mode or transparent mode; you cannot run some contexts in one mode and others in another.

Multiple context mode supports static routing only. For more information about multiple context mode, see Chapter 4, “Enabling Multiple Context Mode.”



CHAPTER 2

Getting Started

This chapter describes how to access the command-line interface, configure the firewall mode, and work with the configuration. This chapter includes the following sections:

- Getting Started with Your Platform Model, page 2-1
- Factory Default Configurations, page 2-1
- Accessing the Command-Line Interface, page 2-4
- Setting Transparent or Routed Firewall Mode, page 2-5
- Working with the Configuration, page 2-6

Getting Started with Your Platform Model

This guide applies to multiple security appliance platforms and models: the PIX 500 series security appliances and the ASA 5500 series adaptive security appliances. There are some hardware differences between the PIX and the ASA security appliance. Moreover, the ASA 5505 includes a built-in switch, and requires some special configuration. For these hardware-based differences, the platforms or models supported are noted directly in each section.

Some models do not support all features covered in this guide. For example, the ASA 5505 adaptive security appliance does not support security contexts. This guide might not list each supported model when discussing a feature. To determine the features that are supported for your model before you start your configuration, see the "Supported Feature Licenses Per Model" section on page 3-1 for a detailed list of the features supported for each model.

Factory Default Configurations

The factory default configuration is the configuration applied by Cisco to new security appliances. The factory default configuration is supported on all models except for the PIX 525 and PIX 535 security appliances.

For the PIX 515/515E and the ASA 5510 and higher security appliances, the factory default configuration configures an interface for management so you can connect to it using ASDM, with which you can then complete your configuration.

For the ASA 5505 adaptive security appliance, the factory default configuration configures interfaces and NAT so that the security appliance is ready to use in your network immediately.

The factory default configuration is available only for routed firewall mode and single context mode. See Chapter 4, "Enabling Multiple Context Mode," for more information about multiple context mode. See the "Setting Transparent or Routed Firewall Mode" section on page 2-5 for more information about routed and transparent firewall mode.

This section includes the following topics:

- Restoring the Factory Default Configuration, page 2-2
- ASA 5505 Default Configuration, page 2-2
- ASA 5510 and Higher Default Configuration, page 2-3
- PIX 515/515E Default Configuration, page 2-4

Restoring the Factory Default Configuration

To restore the factory default configuration, enter the following command:

```
hostname(config)# configure factory-default [ip_address [mask]]
```

If you specify the *ip_address*, then you set the inside or management interface IP address, depending on your model, instead of using the default IP address of 192.168.1.1. The **http** command uses the subnet you specify. Similarly, the **dhcpd address** command range consists of addresses within the subnet that you specify.

After you restore the factory default configuration, save it to internal Flash memory using the **write memory** command. The **write memory** command saves the running configuration to the default location for the startup configuration, even if you previously configured the **boot config** command to set a different location; when the configuration was cleared, this path was also cleared.

Note This command also clears the **boot system** command, if present, along with the rest of the configuration. The **boot system** command lets you boot from a specific image, including an image on the external Flash memory card. The next time you reload the security appliance after restoring the factory configuration, it boots from the first image in internal Flash memory; if you do not have an image in internal Flash memory, the security appliance does not boot.

To configure additional settings that are useful for a full configuration, see the **setup** command.

ASA 5505 Default Configuration

The default factory configuration for the ASA 5505 adaptive security appliance configures the following:

- An inside VLAN 1 interface that includes the Ethernet 0/1 through 0/7 switch ports. If you did not set the IP address in the **configure factory-default** command, then the VLAN 1 IP address and mask are 192.168.1.1 and 255.255.255.0.
- An outside VLAN 2 interface that includes the Ethernet 0/0 switch port. VLAN 2 derives its IP address using DHCP.
- The default route is also derived from DHCP.
- All inside IP addresses are translated when accessing the outside using interface PAT.
- By default, inside users can access the outside, and outside users are prevented from accessing the inside.

- The DHCP server is enabled on the security appliance, so a PC connecting to the VLAN 1 interface receives an address between 192.168.1.2 and 192.168.1.254.
- The HTTP server is enabled for ASDM and is accessible to users on the 192.168.1.0 network.

The configuration consists of the following commands:

```
interface Ethernet 0/0
   switchport access vlan 2
   no shutdown
interface Ethernet 0/1
   switchport access vlan 1
   no shutdown
interface Ethernet 0/2
   switchport access vlan 1
   no shutdown
interface Ethernet 0/3
   switchport access vlan 1
   no shutdown
interface Ethernet 0/4
   switchport access vlan 1
   no shutdown
interface Ethernet 0/5
   switchport access vlan 1
   no shutdown
interface Ethernet 0/6
   switchport access vlan 1
   no shutdown
interface Ethernet 0/7
   switchport access vlan 1
   no shutdown
interface vlan2
   nameif outside
   no shutdown
   ip address dhcp setroute
interface vlan1
   nameif inside
   ip address 192.168.1.1 255.255.255.0
   security-level 100
   no shutdown
global (outside) 1 interface
nat (inside) 1 0 0
http server enable
http 192.168.1.0 255.255.255.0 inside
dhcpd address 192.168.1.2-192.168.1.254 inside
dhcpd auto_config outside
dhcpd enable inside
logging asdm informational
```

ASA 5510 and Higher Default Configuration

The default factory configuration for the ASA 5510 and higher adaptive security appliance configures the following:

- The management interface, Management 0/0. If you did not set the IP address in the **configure factory-default** command, then the IP address and mask are 192.168.1.1 and 255.255.255.0.
- The DHCP server is enabled on the security appliance, so a PC connecting to the interface receives an address between 192.168.1.2 and 192.168.1.254.
- The HTTP server is enabled for ASDM and is accessible to users on the 192.168.1.0 network.

The configuration consists of the following commands:

```
interface management 0/0
   ip address 192.168.1.1 255.255.255.0
   nameif management
   security-level 100
   no shutdown
asdm logging informational 100
asdm history enable
http server enable
http 192.168.1.0 255.255.255.0 management
dhcpd address 192.168.1.2-192.168.1.254 management
dhcpd lease 3600
dhcpd ping_timeout 750
dhcpd enable management
```

PIX 515/515E Default Configuration

The default factory configuration for the PIX 515/515E security appliance configures the following:

- The inside Ethernet1 interface. If you did not set the IP address in the **configure factory-default** command, then the IP address and mask are 192.168.1.1 and 255.255.255.0.
- The DHCP server is enabled on the security appliance, so a PC connecting to the interface receives an address between 192.168.1.2 and 192.168.1.254.
- The HTTP server is enabled for ASDM and is accessible to users on the 192.168.1.0 network.

The configuration consists of the following commands:

```
interface ethernet 1
   ip address 192.168.1.1 255.255.255.0
   nameif management
   security-level 100
   no shutdown
asdm logging informational 100
asdm history enable
http server enable
http 192.168.1.0 255.255.255.0 management
dhcpd address 192.168.1.2-192.168.1.254 management
dhcpd lease 3600
dhcpd ping_timeout 750
dhcpd enable management
```

Accessing the Command-Line Interface

For initial configuration, access the command-line interface directly from the console port. Later, you can configure remote access using Telnet or SSH according to Chapter 42, "Managing System Access." If your system is already in multiple context mode, then accessing the console port places you in the system execution space. See Chapter 4, "Enabling Multiple Context Mode," for more information about multiple context mode.

Note If you want to use ASDM to configure the security appliance instead of the command-line interface, you can connect to the default management address of 192.168.1.1 (if your security appliance includes a factory default configuration. See the "Factory Default Configurations" section on page 2-1.). On the

ASA 5510 and higher adaptive security appliances, the interface to which you connect with ASDM is Management 0/0. For the ASA 5505 adaptive security appliance, the switch port to which you connect with ASDM is any port, except for Ethernet 0/0. For the PIX 515/515E security appliance, the interface to which you connect with ASDM is Ethernet 1. If you do not have a factory default configuration, follow the steps in this section to access the command-line interface. You can then configure the minimum parameters to access ASDM by entering the **setup** command.

To access the command-line interface, perform the following steps:

Step 1 Connect a PC to the console port using the provided console cable, and connect to the console using a terminal emulator set for 9600 baud, 8 data bits, no parity, 1 stop bit, no flow control.

See the hardware guide that came with your security appliance for more information about the console cable.

Step 2 Press the **Enter** key to see the following prompt:

hostname>

This prompt indicates that you are in user EXEC mode.

Step 3 To access privileged EXEC mode, enter the following command:

```
hostname> enable
```

The following prompt appears:

```
Password:
```

Step 4 Enter the enable password at the prompt.

By default, the password is blank, and you can press the **Enter** key to continue. See the "Changing the Enable Password" section on page 9-1 to change the enable password.

The prompt changes to:

```
hostname#
```

To exit privileged mode, enter the **disable**, **exit**, or **quit** command.

Step 5 To access global configuration mode, enter the following command:

```
hostname# configure terminal
```

The prompt changes to the following:

```
hostname(config)#
```

To exit global configuration mode, enter the **exit**, **quit**, or **end** command.

Setting Transparent or Routed Firewall Mode

You can set the security appliance to run in routed firewall mode (the default) or transparent firewall mode.

For multiple context mode, you can use only one firewall mode for all contexts. You must set the mode in the system execution space.

When you change modes, the security appliance clears the configuration because many commands are not supported for both modes. If you already have a populated configuration, be sure to back up your configuration before changing the mode; you can use this backup for reference when creating your new configuration. See the "Backing Up Configuration Files" section on page 43-8. For multiple context mode, the system configuration is erased. This action removes any contexts from running. If you then re-add a context that has an existing configuration that was created for the wrong mode, the context configuration will not work correctly. Be sure to recreate your context configurations for the correct mode before you re-add them, or add new contexts with new paths for the new configurations.

If you download a text configuration to the security appliance that changes the mode with the **firewall transparent** command, be sure to put the command at the top of the configuration; the security appliance changes the mode as soon as it reads the command and then continues reading the configuration you downloaded. If the command is later in the configuration, the security appliance clears all the preceding lines in the configuration. See the "Downloading Software or Configuration Files to Flash Memory" section on page 43-2 for information about downloading text files.

- To set the mode to transparent, enter the following command in the system execution space:

  ```
  hostname(config)# firewall transparent
  ```

  This command also appears in each context configuration for informational purposes only; you cannot enter this command in a context.

- To set the mode to routed, enter the following command in the system execution space:

  ```
  hostname(config)# no firewall transparent
  ```

Working with the Configuration

This section describes how to work with the configuration. The security appliance loads the configuration from a text file, called the startup configuration. This file resides by default as a hidden file in internal Flash memory. You can, however, specify a different path for the startup configuration. (For more information, see Chapter 43, "Managing Software and Configurations.")

When you enter a command, the change is made only to the running configuration in memory. You must manually save the running configuration to the startup configuration for your changes to remain after a reboot.

The information in this section applies to both single and multiple security contexts, except where noted. Additional information about contexts is in Chapter 4, "Enabling Multiple Context Mode."

This section includes the following topics:

- Saving Configuration Changes, page 2-6
- Copying the Startup Configuration to the Running Configuration, page 2-8
- Viewing the Configuration, page 2-8
- Clearing and Removing Configuration Settings, page 2-9
- Creating Text Configuration Files Offline, page 2-9

Saving Configuration Changes

This section describes how to save your configuration, and includes the following topics:

- Saving Configuration Changes in Single Context Mode, page 2-7

- Saving Configuration Changes in Multiple Context Mode, page 2-7

Saving Configuration Changes in Single Context Mode

To save the running configuration to the startup configuration, enter the following command:

```
hostname# write memory
```

Note The **copy running-config startup-config** command is equivalent to the **write memory** command.

Saving Configuration Changes in Multiple Context Mode

You can save each context (and system) configuration separately, or you can save all context configurations at the same time. This section includes the following topics:

- Saving Each Context and System Separately, page 2-7
- Saving All Context Configurations at the Same Time, page 2-7

Saving Each Context and System Separately

To save the system or context configuration, enter the following command within the system or context:

```
hostname# write memory
```

Note The **copy running-config startup-config** command is equivalent to the **write memory** command.

For multiple context mode, context startup configurations can reside on external servers. In this case, the security appliance saves the configuration back to the server you identified in the context URL, except for an HTTP or HTTPS URL, which do not let you save the configuration to the server.

Saving All Context Configurations at the Same Time

To save all context configurations at the same time, as well as the system configuration, enter the following command in the system execution space:

```
hostname# write memory all [/noconfirm]
```

If you do not enter the **/noconfirm** keyword, you see the following prompt:

```
Are you sure [Y/N]:
```

After you enter **Y**, the security appliance saves the system configuration and each context. Context startup configurations can reside on external servers. In this case, the security appliance saves the configuration back to the server you identified in the context URL, except for an HTTP or HTTPS URL, which do not let you save the configuration to the server.

After the security appliance saves each context, the following message appears:

```
'Saving context 'b' ... ( 1/3 contexts saved ) '
```

Sometimes, a context is not saved because of an error. See the following information for errors:

- For contexts that are not saved because of low memory, the following message appears:

  ```
  The context 'context a' could not be saved due to Unavailability of resources
  ```

- For contexts that are not saved because the remote destination is unreachable, the following message appears:

  ```
  The context 'context a' could not be saved due to non-reachability of destination
  ```

- For contexts that are not saved because the context is locked, the following message appears:

  ```
  Unable to save the configuration for the following contexts as these contexts are
  locked.
  context ‘a’ , context ‘x’ , context ‘z’ .
  ```

  A context is only locked if another user is already saving the configuration or in the process of deleting the context.

- For contexts that are not saved because the startup configuration is read-only (for example, on an HTTP server), the following message report is printed at the end of all other messages:

  ```
  Unable to save the configuration for the following contexts as these contexts have
  read-only config-urls:
  context ‘a’ , context ‘b’ , context ‘c’ .
  ```

- For contexts that are not saved because of bad sectors in the Flash memory, the following message appears:

  ```
  The context 'context a' could not be saved due to Unknown errors
  ```

Copying the Startup Configuration to the Running Configuration

Copy a new startup configuration to the running configuration using one of these options:

- To merge the startup configuration with the running configuration, enter the following command:

  ```
  hostname(config)# copy startup-config running-config
  ```

  A merge adds any new commands from the new configuration to the running configuration. If the configurations are the same, no changes occur. If commands conflict or if commands affect the running of the context, then the effect of the merge depends on the command. You might get errors, or you might have unexpected results.

- To load the startup configuration and discard the running configuration, restart the security appliance by entering the following command:

  ```
  hostname# reload
  ```

  Alternatively, you can use the following commands to load the startup configuration and discard the running configuration without requiring a reboot:

  ```
  hostname/contexta(config)# clear configure all
  hostname/contexta(config)# copy startup-config running-config
  ```

Viewing the Configuration

The following commands let you view the running and startup configurations.

- To view the running configuration, enter the following command:

  ```
  hostname# show running-config
  ```

- To view the running configuration of a specific command, enter the following command:

  ```
  hostname# show running-config command
  ```

- To view the startup configuration, enter the following command:

  ```
  hostname# show startup-config
  ```

Clearing and Removing Configuration Settings

To erase settings, enter one of the following commands.

- To clear all the configuration for a specified command, enter the following command:

  ```
  hostname(config)# clear configure configurationcommand [level2configurationcommand]
  ```

  This command clears all the current configuration for the specified configuration command. If you only want to clear the configuration for a specific version of the command, you can enter a value for *level2configurationcommand*.

  For example, to clear the configuration for all **aaa** commands, enter the following command:

  ```
  hostname(config)# clear configure aaa
  ```

  To clear the configuration for only **aaa authentication** commands, enter the following command:

  ```
  hostname(config)# clear configure aaa authentication
  ```

- To disable the specific parameters or options of a command, enter the following command:

  ```
  hostname(config)# no configurationcommand [level2configurationcommand] qualifier
  ```

  In this case, you use the **no** command to remove the specific configuration identified by *qualifier*.

  For example, to remove a specific **nat** command, enter enough of the command to identify it uniquely as follows:

  ```
  hostname(config)# no nat (inside) 1
  ```

- To erase the startup configuration, enter the following command:

  ```
  hostname(config)# write erase
  ```

- To erase the running configuration, enter the following command:

  ```
  hostname(config)# clear configure all
  ```

Note In multiple context mode, if you enter **clear configure all** from the system configuration, you also remove all contexts and stop them from running.

Creating Text Configuration Files Offline

This guide describes how to use the CLI to configure the security appliance; when you save commands, the changes are written to a text file. Instead of using the CLI, however, you can edit a text file directly on your PC and paste a configuration at the configuration mode command-line prompt in its entirety, or line by line. Alternatively, you can download a text file to the security appliance internal Flash memory. See Chapter 43, "Managing Software and Configurations," for information on downloading the configuration file to the security appliance.

In most cases, commands described in this guide are preceded by a CLI prompt. The prompt in the following example is "hostname(config)#":

```
hostname(config)# context a
```

In the text configuration file you are not prompted to enter commands, so the prompt is omitted as follows:

```
context a
```

For additional information about formatting the file, see Appendix B, "Using the Command-Line Interface."



CHAPTER 3

Managing Feature Licenses

A license specifies the options that are enabled on a given security appliance. It is represented by an activation key which is a 160-bit (5 32-bit words or 20 bytes) value. This value encodes the serial number (an 11 character string) and the enabled features.

This chapter describes how to obtain an activation key and activate it. It also describes the available licenses for each model. This chapter includes the following sections:

This document includes the following sections:

- Supported Feature Licenses Per Model, page 3-1
- Information About Feature Licenses, page 3-9
- Guidelines and Limitations, page 3-12
- Viewing Your Current License, page 3-12
- Obtaining an Activation Key, page 3-14
- Entering a New Activation Key, page 3-15
- Upgrading the License for a Failover Pair, page 3-16
- Feature History for Licensing, page 3-18

Supported Feature Licenses Per Model

This section lists the feature licenses available for each model:

- ASA 5505, Table 3-1 on page 3-2
- ASA 5510, Table 3-2 on page 3-3
- ASA 5520, Table 3-3 on page 3-4
- ASA 5540, Table 3-4 on page 3-5
- ASA 5550, Table 3-5 on page 3-6
- PIX 515/515E, Table 3-6 on page 3-7
- PIX 525. Table 3-7 on page 3-8
- PIX 535, Table 3-8 on page 3-9

Note The ASA 5580 is not supported in Version 8.0; for ASA 5580 information, see the licensing documentation for Version 8.1 or later.

The PIX 500 series security appliance does not support temporary licenses.

Items that are in italics are separate, optional licenses with which that you can replace the Base or Security Plus license. You can mix and match licenses, for example, the 10 security context license plus the Strong Encryption license; or the 500 SSL VPN license plus the GTP/GPRS license; or all four licenses together.

Table 3-1 ASA 5505 Adaptive Security Appliance License Features

ASA 5505	Base License		Security Plus	
Users, concurrent[1]	10[2]	*Optional licenses:*	10[2]	*Optional licenses:*
		50 *Unlimited*		*50* *Unlimited*
Security Contexts	No support		No support	
VPN Sessions[3]	25 combined IPSec and SSL VPN		25 combined IPSec and SSL VPN	
Max. IPSec Sessions	10		25	
Max. SSL VPN Sessions	2	*Optional licenses:*	2	*Optional licenses:*
		10 *25*		*10* *25*
VPN Load Balancing	No support		No support	
Advanced Endpoint Assessment	None	*Optional license: Enabled*	None	*Optional license: Enabled*
Unified Communications Proxy Sessions[4]	2	*Optional license: 24*	2	*Optional license: 24*
Failover	No support		Active/Standby (no stateful failover)	
GTP/GPRS	No support		No support	
Maximum VLANs/Zones	3 (2 regular zones and 1 restricted zone that can only communicate with 1 other zone)		20	
Maximum VLAN Trunks	No support		8 trunks	
Concurrent Firewall Conns	10 K		25 K	
Max. Physical Interfaces	Unlimited, assigned to VLANs/zones		Unlimited, assigned to VLANs/zones	
Encryption	Base (DES)	*Optional license: Strong (3DES/AES)*	Base (DES)	*Optional license: Strong (3DES/AES)*
Minimum RAM	256 MB (default)		256 MB (default)	

1. In routed mode, hosts on the inside (Business and Home VLANs) count towards the limit only when they communicate with the outside (Internet VLAN). Internet hosts are not counted towards the limit. Hosts that initiate traffic between Business and Home are also not counted towards the limit. The interface associated with the default route is considered to be the Internet interface. If there is no default route, hosts on all interfaces are counted toward the limit. In transparent mode, the interface with the lowest number of hosts is counted towards the host limit. See the **show local-host** command to view the host limits.
2. For a 10-user license, the max. DHCP clients is 32. For 50 users, the max. is 128. For unlimited users, the max. is 250, which is the max. for other models.
3. Although the maximum IPSec and SSL VPN sessions add up to more than the maximum VPN sessions, the combined sessions should not exceed the VPN session limit. If you exceed the maximum VPN sessions, you can overload the security appliance, so be sure to size your network appropriately. When determining the session makeup of the combined limit, the number of SSL VPN sessions cannot exceed the number of licensed SSL VPN sessions on the security appliance (which is 2 by default).
4. Phone Proxy, Mobility Proxy, Presence Federation Proxy, and TLS Proxy are all licensed under the UC Proxy umbrella, and can be mixed and matched. For example, if you configure a phone with a primary and backup Cisco Unified Communications Manager, there are 2 TLS/SRTP connections, and 2 UC Proxy sessions are used. This license was introduced in Version 8.0(4). In prior versions, TLS proxy for SIP and Skinny inspection was included in the Base License.

Table 3-2 ASA 5510 Adaptive Security Appliance License Features

ASA 5510	Base License	Security Plus
Users, concurrent	Unlimited	Unlimited
Security Contexts	No support	2 *Optional licenses:* *5*
VPN Sessions[1]	250 combined IPSec and SSL VPN	250 combined IPSec and SSL VPN
Max. IPSec Sessions	250	250
Max. SSL VPN Sessions	2 *Optional licenses:* *10* *25* *50* *100* *250* *Optional VPN Flex license:[2] 250*	2 *Optional licenses:* *10* *25* *50* *100* *250* *Optional VPN Flex license: 250*
VPN Load Balancing	No support	Supported
Advanced Endpoint Assessment	None *Optional license: Enabled*	None *Optional license: Enabled*
Unified Communications Proxy Sessions[3] (introduced in 8.0(4))	2 *Optional licenses* *24* *50* *100*	2 *Optional licenses* *24* *50* *100*
Failover	No support	Active/Standby or Active/Active[4]
GTP/GPRS	No support	No support
Max. VLANs	50	100
Concurrent Firewall Conns	50 K	130 K
Max. Physical Interfaces	Unlimited	Unlimited
Encryption	Base (DES) *Optional license: Strong (3DES/AES)*	Base (DES) *Optional license: Strong (3DES/AES)*
Min. RAM	256 MB (default)	256 MB (default)

1. Although the maximum IPSec and SSL VPN sessions add up to more than the maximum VPN sessions, the combined sessions should not exceed the VPN session limit. If you exceed the maximum VPN sessions, you can overload the security appliance, so be sure to size your network appropriately. When determining the session makeup of the combined limit, the number of SSL VPN sessions cannot exceed the number of licensed SSL VPN sessions on the security appliance (which is 2 by default).
2. Available in Version 8.0(4) and later.
3. Phone Proxy, Mobility Proxy, Presence Federation Proxy, and TLS Proxy are all licensed under the UC Proxy umbrella, and can be mixed and matched. For example, if you configure a phone with a primary and backup Cisco Unified Communications Manager, there are 2 TLS/SRTP connections, and 2 UC Proxy sessions are used. This license was introduced in Version 8.0(4). In prior versions, TLS proxy for SIP and Skinny inspection was included in the Base License.
4. You cannot use Active/Active failover and VPN; if you want to use VPN, use Active/Standby failover.

Table 3-3 ASA 5520 Adaptive Security Appliance License Features

ASA 5520	Base License						
Users, concurrent	Unlimited					Unlimited	
Security Contexts	2	*Optional licenses:*					
		5	*10*	*20*			
VPN Sessions[1]	750 combined IPSec and SSL VPN						
Max. IPSec Sessions	750						
Max. SSL VPN Sessions	2	*Optional licenses:*					
		10	*25*	*50*	*100*	*250*	*500*	*750*
		Optional VPN Flex licenses:[2]				*250*		*750*
VPN Load Balancing	Supported						
Advanced Endpoint Assessment	None	*Optional license: Enabled*						
Unified Communications Proxy Sessions[3]	2	*Optional licenses*						
		24	*50*	*100*	*250*	*500*	*750*	*1000*
Failover	Active/Standby or Active/Active[4]						
GTP/GPRS	None	*Optional license: Enabled*						
Max. VLANs	150						
Concurrent Firewall Conns	280 K						
Max. Physical Interfaces	Unlimited						
Encryption	Base (DES)	*Optional license: Strong (3DES/AES)*						
Min. RAM	512 MB (default)						

1. Although the maximum IPSec and SSL VPN sessions add up to more than the maximum VPN sessions, the combined sessions should not exceed the VPN session limit. If you exceed the maximum VPN sessions, you can overload the security appliance, so be sure to size your network appropriately. When determining the session makeup of the combined limit, the number of SSL VPN sessions cannot exceed the number of licensed SSL VPN sessions on the security appliance (which is 2 by default).
2. Available in Version 8.0(4) and later.
3. Phone Proxy, Mobility Proxy, Presence Federation Proxy, and TLS Proxy are all licensed under the UC Proxy umbrella, and can be mixed and matched. For example, if you configure a phone with a primary and backup Cisco Unified Communications Manager, there are 2 TLS/SRTP connections, and 2 UC Proxy sessions are used. This license was introduced in Version 8.0(4). In prior versions, TLS proxy for SIP and Skinny inspection was included in the Base License.
4. You cannot use Active/Active failover and VPN; if you want to use VPN, use Active/Standby failover.

Table 3-4 ASA 5540 Adaptive Security Appliance License Features

ASA 5540	Base License								
Users, concurrent	Unlimited					Unlimited			
Security Contexts	2	*Optional licenses:*							
		5	*10*	*20*	*50*				
VPN Sessions[1]	5000 combined IPSec and SSL VPN								
Max. IPSec Sessions	5000								
Max. SSL VPN Sessions	2	*Optional licenses:*							
		10	*25*	*50*	*100*	*250*	*500*	*750*	*1000*	*2500*
		Optional VPN Flex Licenses:[2]				*250*		*750*	*1000*	*2500*
VPN Load Balancing	Supported								
Advanced Endpoint Assessment	None	*Optional license: Enabled*							
Unified Communications Proxy Sessions[3]	2	*Optional licenses*							
		24	*50*	*100*	*250*	*500*	*750*	*1000*	*2000*
Failover	Active/Standby or Active/Active[4]								
GTP/GPRS	None	*Optional license: Enabled*							
Max. VLANs	200								
Concurrent Firewall Conns	400 K								
Max. Physical Interfaces	Unlimited								
Encryption	Base (DES)	*Optional license: Strong (3DES/AES)*							
Min. RAM	1 GB (default)								

1. Although the maximum IPSec and SSL VPN sessions add up to more than the maximum VPN sessions, the combined sessions should not exceed the VPN session limit. If you exceed the maximum VPN sessions, you can overload the security appliance, so be sure to size your network appropriately. When determining the session makeup of the combined limit, the number of SSL VPN sessions cannot exceed the number of licensed SSL VPN sessions on the security appliance (which is 2 by default). This license was introduced in Version 8.0(4). In prior versions, TLS proxy for SIP and Skinny inspection was included in the Base License.
2. Available in Version 8.0(4) and later.
3. Phone Proxy, Mobility Proxy, Presence Federation Proxy, and TLS Proxy are all licensed under the UC Proxy umbrella, and can be mixed and matched. For example, if you configure a phone with a primary and backup Cisco Unified Communications Manager, there are 2 TLS/SRTP connections, and 2 UC Proxy sessions are used. Prior to 8.0(4), only TLS Proxy was available.
4. You cannot use Active/Active failover and VPN; if you want to use VPN, use Active/Standby failover.

Table 3-5 ASA 5550 Adaptive Security Appliance License Features

ASA 5550	Base License									
Users, concurrent	Unlimited									
Security Contexts	2	*Optional licenses:*								
		5	*10*	*20*	*50*					
VPN Sessions[1]	5000 combined IPSec and SSL VPN									
Max. IPSec Sessions	5000									
Max. SSL VPN Sessions	2	*Optional licenses:*								
		10	*25*	*50*	*100*	*250*	*500*	*750*	*1000*	*2500*	*5000*
		Optional VPN Flex licenses:[2]				*250*		*750*	*1000*	*2500*	*5000*
VPN Load Balancing	Supported									
Advanced Endpoint Assessment	None	*Optional license: Enabled*								
Unified Communications Proxy Sessions[3]	2	*Optional licenses*								
		24	*50*	*100*	*250*	*500*	*750*	*1000*	*2000*	*3000*
Failover	Active/Standby or Active/Active[4]									
GTP/GPRS	None	*Optional license: Enabled*								
Max. VLANs	250									
Concurrent Firewall Conns	650 K									
Max. Physical Interfaces	Unlimited									
Encryption	Base (DES)	*Optional license: Strong (3DES/AES)*								
Min. RAM	4 GB (default)									

1. Although the maximum IPSec and SSL VPN sessions add up to more than the maximum VPN sessions, the combined sessions should not exceed the VPN session limit. If you exceed the maximum VPN sessions, you can overload the security appliance, so be sure to size your network appropriately. When determining the session makeup of the combined limit, the number of SSL VPN sessions cannot exceed the number of licensed SSL VPN sessions on the security appliance (which is 2 by default). This license was introduced in Version 8.0(4). In prior versions, TLS proxy for SIP and Skinny inspection was included in the Base License.
2. Available in Version 8.0(4) and later.
3. Phone Proxy, Mobility Proxy, Presence Federation Proxy, and TLS Proxy are all licensed under the UC Proxy umbrella, and can be mixed and matched. For example, if you configure a phone with a primary and backup Cisco Unified Communications Manager, there are 2 TLS/SRTP connections, and 2 UC Proxy sessions are used.
4. You cannot use Active/Active failover and VPN; if you want to use VPN, use Active/Standby failover.

Table 3-6 PIX 515/515E Security Appliance License Features

PIX 515/515E	R (Restricted)			UR (Unrestricted)			FO (Failover)[1]			FO-AA (Failover Active/Active)[1]		
Advanced Endpoint Assessment	No support			No support			No support			No support		
Encryption	None	*Optional licenses:*		None	*Optional licenses:*		None	*Optional licenses:*		None	*Optional licenses:*	
		Base (DES)	*Strong (3DES/ AES)*		*Base (DES)*	*Strong (3DES/ AES)*		*Base (DES)*	*Strong (3DES/ AES)*		*Base (DES)*	*Strong (3DES/ AES)*
Failover	No support			Active/Standby Active/Active			Active/Standby			Active/Standby Active/Active		
Firewall Conns, concurrent	48 K			130 K			130 K			130 K		
GTP/GPRS	None	*Optional license: Enabled*		None	*Optional license: Enabled*		None	*Optional license: Enabled*		None	*Optional license: Enabled*	
IPSec Sessions	2000			2000			2000			2000		
Physical Interfaces, max.	3			6			6			6		
RAM, min.	64 MB (default)			128 MB			128 MB			128 MB		
Security Contexts	No support			2	*Optional license: 5*		2	*Optional license: 5*		2	*Optional license: 5*	
SSL VPN Sessions	No support			No support			No support			No support		
Unified Communications Proxy Sessions	No support			No support			No support			No support		
Users, concurrent	Unlimited			Unlimited			Unlimited			Unlimited		
VLANs, max.	10			25			25			25		
VPN Load Balancing	No support			No support			No support			No support		

1. This license can only be used in a failover pair with another unit with a UR license. Both units must be the same model.

Table 3-7 PIX 525 Security Appliance License Features

PIX 525	R (Restricted)			UR (Unrestricted)			FO (Failover)[1]			FO-AA (Failover Active/Active)[1]		
Advanced Endpoint Assessment	No support			No support			No support			No support		
Encryption	None	*Optional licenses:*		None	*Optional licenses:*		None	*Optional licenses:*		None	*Optional licenses:*	
		Base (DES)	*Strong (3DES/ AES)*		*Base (DES)*	*Strong (3DES/ AES)*		*Base (DES)*	*Strong (3DES/ AES)*		*Base (DES)*	*Strong (3DES/ AES)*
Failover	No support			Active/Standby Active/Active			Active/Standby			Active/Standby Active/Active		
Firewall Conns, concurrent	140 K			280 K			280 K			280 K		
GTP/GPRS	None	*Optional license: Enabled*		None	*Optional license: Enabled*		None	*Optional license: Enabled*		None	*Optional license: Enabled*	
IPSec Sessions	2000			2000			2000			2000		
Physical Interfaces, max.	6			10			10			10		
RAM, min.	128 MB (default)			256 MB			256 MB			256 MB		
Security Contexts	No support			2	*Optional licenses:*		2	*Optional licenses:*		2	*Optional licenses:*	
				5	*10*	*20*	*50*	*5*	*10*	*20*	*50*	*5*	*10*	*20*	*50*
SSL VPN Sessions	No support			No support			No support			No support		
Unified Communications Proxy Sessions	No support			No support			No support			No support		
Users, concurrent	Unlimited			Unlimited			Unlimited			Unlimited		
VLANs, max.	25			100			100			100		
VPN Load Balancing	No support			No support			No support			No support		

1. This license can only be used in a failover pair with another unit with a UR license. Both units must be the same model.

Table 3-8 PIX 535 Security Appliance License Features

PIX 535	R (Restricted)			UR (Unrestricted)			FO (Failover)[1]			FO-AA (Failover Active/Active)[1]		
Advanced Endpoint Assessment	No support			No support			No support			No support		
Encryption	None	*Optional licenses:*		None	*Optional licenses:*		None	*Optional licenses:*		None	*Optional licenses:*	
		Base (DES)	*Strong (3DES/ AES)*		*Base (DES)*	*Strong (3DES/ AES)*		*Base (DES)*	*Strong (3DES/ AES)*		*Base (DES)*	*Strong (3DES/ AES)*
Failover	No support			Active/Standby Active/Active			Active/Standby			Active/Standby Active/Active		
Firewall Conns, concurrent	250 K			500 K			500 K			500 K		
GTP/GPRS	None	*Optional license: Enabled*		None	*Optional license: Enabled*		None	*Optional license: Enabled*		None	*Optional license: Enabled*	
IPSec Sessions	2000			2000			2000			2000		
Physical Interfaces, max.	8			14			14			14		
RAM, min.	512 MB (default)			1024 MB			1024 MB			1024 MB		
Security Contexts	No support			2	*Optional licenses:*		2	*Optional licenses:*		2	*Optional licenses:*	
				5	*10*	*20*	*50*	*5*	*10*	*20*	*50*	*5*	*10*	*20*	*50*
SSL VPN Sessions	No support			No support			No support			No support		
Unified Communications Proxy Sessions	No support			No support			No support			No support		
Users, concurrent	Unlimited			Unlimited			Unlimited			Unlimited		
VLANs, max.	50			150			150			150		
VPN Load Balancing	No support			No support			No support			No support		

1. This license can only be used in a failover pair with another unit with a UR license. Both units must be the same model.

Information About Feature Licenses

A license specifies the options that are enabled on a given security appliance. It is represented by an activation key which is a 160-bit (5 32-bit words or 20 bytes) value. This value encodes the serial number (an 11 character string) and the enabled features.

This section includes the following topics:

- Preinstalled License, page 3-10
- VPN Flex and Evaluation Licenses, page 3-10

Preinstalled License

By default, your security appliance ships with a license already installed. This license might be the Base License, to which you want to add more licenses, or it might already have all of your licenses installed, depending on what you ordered and what your vendor installed for you. See the "Viewing Your Current License" section on page 3-12 section to determine which licenses you have installed.

VPN Flex and Evaluation Licenses

Note The PIX 500 series security appliance does not support temporary licenses.

In addition to permanent licenses, you can purchase a temporary VPN Flex license or receive an evaluation license that has a time-limit. For example, you might buy a VPN Flex license to handle short-term surges in the number of concurrent SSL VPN users.

This section includes the following topics:

- How the Temporary License Timer Works, page 3-10
- How Multiple Licenses Interact, page 3-11
- Failover and Temporary Licenses, page 3-11

How the Temporary License Timer Works

- The timer for the temporary license starts counting down when you activate it on the security appliance.
- If you stop using the temporary license before it times out, for example you activate a permanent license or a different temporary license, then the timer halts. The timer only starts again when you reactivate the temporary license.
- If the temporary license is active, and you shut down the security appliance, then the timer continues to count down. If you intend to leave the security appliance in a shut down state for an extended period of time, then you should activate the permanent license before you shut down to preserve the temporary license.
- When a temporary license expires, the next time you reload the security appliance, the permanent license is used; you are not forced to perform a reload immediately when the license expires.

Note We suggest you do not change the system clock after you install the temporary license. If you set the clock to be a later date, then if you reload, the security appliance checks the system clock against the original installation time, and assumes that more time has passed than has actually been used. If you set the clock back, and the actual running time is greater than the time between the original installation time and the system clock, then the license immediately expires after a reload.

How Multiple Licenses Interact

- When you activate a temporary license, then features from both permanent and temporary licenses combine to form the running license. The security appliance uses the highest value from each license for each feature, and displays any resolved conflicts between the licenses when you enter a temporary activation key. In the rare circumstance that a temporary license has lower capability than the permanent license, the permanent license values are used.
- When you activate a permanent license, it overwrites the currently-running permanent and temporary licenses and becomes the running license.

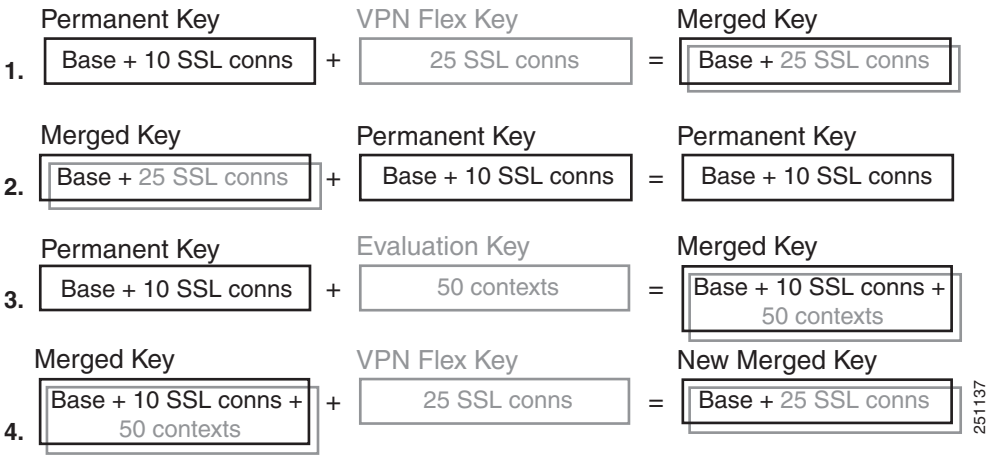
Note If the permanent license is a *downgrade* from the temporary license, then you need to reload the security appliance to disable the temporary license and restore the permanent license. Until you reload, the temporary license continues to count down.

Interim release 8.0(4.16) includes an enhancement so that you do not need to reload the security appliance after reactivating the already installed permanent license; this enhancement stops the temporary license from continuing to count down with no disruption of traffic.

- To reenable the features of the temporary license if you later activate a permanent license, simply reenter the temporary activation key. For a license upgrade, you do not need to reload.
- To switch to a different temporary license, enter the new activation key; the new license is used instead of the old temporary license and combines with the permanent license to create a new running license. The security appliance can have multiple temporary licenses installed; but only one is active at any given time.

See the following figure for examples of permanent and VPN Flex activation keys, and how they interact.

Figure 3-1 Permanent and VPN Flex Activation Keys



Failover and Temporary Licenses

Because the temporary license continues to count down for as long as it is activated on a failover unit, we do not recommend using a temporary license in a failover situation, except in an emergency where the temporary license is activated only for a short period of time. In this case, one unit can use the

permanent license and the other unit can use the temporary license if the features are equivalent between the permanent and temporary licenses. This functionality is useful if the hardware fails on a unit, and you need to replace it for a short period of time until the replacement unit arrives.

Guidelines and Limitations

See the following guidelines for activation keys.

Context Mode Guidelines

In multiple context mode, apply the activation key in the system execution space.

Firewall Mode Guidelines

Activation keys are available in both routed and transparent mode.

Failover Guidelines

Because the temporary license continues to count down for as long as it is activated on a failover unit, we do not recommend using a temporary license in a failover situation, except in an emergency where the temporary license is activated only for a short period of time. In this case, one unit can use the permanent license and the other unit can use the temporary license if the features are equivalent between the permanent and temporary licenses. This functionality is useful if the hardware fails on a unit, and you need to replace it for a short period of time until the replacement unit arrives.

Additional Guidelines and Limitations

- The activation key is not stored in your configuration file; it is stored as a hidden file in Flash memory.
- The activation key is tied to the serial number of the device. Feature licenses cannot be transferred between devices (except in the case of a hardware failure). If you have to replace your device due to a hardware failure, contact the Cisco Licensing Team to have your existing license transferred to the new serial number. The Cisco Licensing Team will ask for the Product Authorization Key reference number and existing serial number.
- Once purchased, you cannot return a license for a refund or for an upgraded license.
- You cannot add two separate licenses for the same feature together; for example, if you purchase a 25-session SSL VPN license, and later purchase a 50-session license, you cannot use 75 sessions; you can use a maximum of 50 sessions.

Viewing Your Current License

This section describes how to view your current license, and for temporary activation keys, how much time the license has left.

Detailed Steps

Command	Purpose
show activation-key detail **Example:** hostname# **show activation-key detail**	Shows the installed licenses, including information about temporary licenses.

Examples

The following is sample output from the **show activation-key detail** command that shows a permanent activation license, an active temporary license, the merged running license, and also the activation keys for inactive temporary licenses:

```
hostname# show activation-key detail

Serial Number:  JMX0916L0Z4

Permanent Flash Activation Key: 0xf412675d 0x48a446bc 0x8c532580 0xb000b8c4 0xcc21f48e

Licensed features for this platform:
Maximum Physical Interfaces  : Unlimited
Maximum VLANs                : 200
Inside Hosts                 : Unlimited
Failover                     : Active/Active
VPN-DES                      : Enabled
VPN-3DES-AES                 : Enabled
Security Contexts            : 2
GTP/GPRS                     : Disabled
VPN Peers                    : 5000
WebVPN Peers                 : 2
AnyConnect for Mobile        : Disabled
AnyConnect for Linksys phone : Disabled
Advanced Endpoint Assessment : Disabled
UC Proxy Sessions:           : 2

Temporary Flash Activation Key: 0xcb0367ce 0x700dd51d 0xd57b98e3 0x6ebcf553 0x0b058aac

Licensed features for this platform:
Maximum Physical Interfaces  : Unlimited
Maximum VLANs                : 200
Inside Hosts                 : Unlimited
Failover                     : Active/Active
VPN-DES                      : Enabled
VPN-3DES-AES                 : Disabled
Security Contexts            : 2
GTP/GPRS                     : Disabled
VPN Peers                    : 5000
WebVPN Peers                 : 500
AnyConnect for Mobile        : Disabled
AnyConnect for Linksys phone : Disabled
Advanced Endpoint Assessment : Disabled
UC Proxy Sessions:           : 2
This is a time-based license that will expire in 27 day(s).

Running Activation Key: 0xcb0367ce 0x700dd51d 0xd57b98e3 0x6ebcf553 0x0b058aac

Licensed features for this platform:
Maximum Physical Interfaces  : Unlimited
Maximum VLANs                : 200
```

```
Inside Hosts                : Unlimited
Failover                    : Active/Active
VPN-DES                     : Enabled
VPN-3DES-AES                : Enabled
Security Contexts           : 2
GTP/GPRS                    : Disabled
VPN Peers                   : 5000
WebVPN Peers                : 500
AnyConnect for Mobile       : Disabled
AnyConnect for Linksys phone : Disabled
Advanced Endpoint Assessment : Disabled
UC Proxy Sessions:          : 2


This platform has an ASA 5540 VPN Premium license.
This is a time-based license that will expire in 27 day(s).

The flash activation key is the SAME as the running key.

Non-active temporary keys:                                Time left
-----------------------------------------------------------------
0x2a53d6   0xfc087bfe 0x691b94fb 0x73dc8bf3 0xcc028ca2  28 day(s)
0xa13a46c2 0x7c10ec8d 0xad8a2257 0x5ec0ab7f 0x86221397  27 day(s)
```

Obtaining an Activation Key

To obtain an activation key, you need a Product Authorization Key, which you can purchase from your Cisco account representative. You need to purchase a separate Product Activation Key for each feature license. For example, if you have the Base License, you can purchase separate keys for Advanced Endpoint Assessment and for additional SSL VPN sessions.

Note For a failover pair, you need separate activation keys for each unit. Make sure the licenses included in the keys are the same for both units.

After obtaining the Product Authorization Keys, register them on Cisco.com by performing the following steps:

Step 1 Obtain the serial number for your security appliance by entering the following command:

```
hostname# show activation-key
```

Step 2 Access one of the following URLs.

- Use the following website if you are a registered user of Cisco.com:

  `http://www.cisco.com/go/license`

- Use the following website if you are not a registered user of Cisco.com:

  `http://www.cisco.com/go/license/public`

Step 3 Enter the following information, when prompted:

- Product Authorization Key (if you have multiple keys, enter one of the keys first. You have to enter each key as a separate process.)
- The serial number of your security appliance

- Your email address

An activation key is automatically generated and sent to the email address that you provide. This key includes all features you have registered so far for permanent licenses. For VPN Flex licenses, each license has a separate activation key.

Step 4 If you have additional Product Authorization Keys, repeat Step 3 for each Product Authorization Key. After you enter all of the Product Authorization Keys, the final activation key provided includes all of the permanent features you registered.

Entering a New Activation Key

Before entering the activation key, ensure that the image in Flash memory and the running image are the same. You can do this by reloading the security appliance before entering the new activation key.

Prerequisites

- If you are already in multiple context mode, enter the activation key in the system execution space.
- Some licenses require you to reload the security appliance after you activate them. Table 3-9 lists the licenses that require reloading.

Table 3-9 License Reloading Requirements

Model	License Action Requiring Reload
ASA 5505 and ASA 5510	Changing between the Base and Security Plus license.
PIX 500 series	Changing between R, UR, FO, and FO-AA licenses.
All models	Changing the Encryption license.
All models	Downgrading any license (for example, going from 10 contexts to 2 contexts). **Note** If a temporary license expires, and the permanent license is a downgrade, then you do not need to immediately reload the security appliance; the next time you reload, the permanent license is restored.

Detailed Steps

	Command	Purpose
Step 1	**activation-key** *key* **Example:** `hostname(config)# activation-key 0xd11b3d48 0xa80a4c0a 0x48e0fd1c 0xb0443480 0x843fc490`	Applies an activation key to the security appliance. The key is a five-element hexadecimal string with one space between each element. The leading 0x specifier is optional; all values are assumed to be hexadecimal. You can enter one permanent key, and multiple temporary keys. The last temporary key entered is the active one. See the "VPN Flex and Evaluation Licenses" section on page 3-10 for more information. To change the running activation key, enter the **activation-key** command with a new key value.
Step 2	**reload** **Example:** `hostname(config)# reload`	(Might be required.) Reloads the security appliance. Some licenses require you to reload the security appliance after entering the new activation key. See Table 3-9 on page 3-15 for a list of licenses that need reloading. If you need to reload, you will see the following message: `WARNING: The running activation key was not updated with the requested key. The flash activation key was updated with the requested key, and will become active after the next reload.`

Upgrading the License for a Failover Pair

If you need to upgrade the license on a failover pair, you might have some amount of downtime depending on whether the license requires a reload. See Table 3-9 on page 3-15 for more information about licenses requiring a reload. This section includes the following topics:

- Upgrading the License for a Failover (No Reload Required), page 3-16
- Upgrading the License for a Failover (Reload Required), page 3-17

Upgrading the License for a Failover (No Reload Required)

Use the following procedure if your new license does not require you to reload. See Table 3-9 on page 3-15 for more information about licenses requiring a reload. This procedure ensures that there is no downtime.

Detailed Steps

	Command	Purpose
	On the active unit:	
Step 1	**no failover** **Example:** `active(config)# no failover`	Disables failover on the active unit. The standby unit remains in standby mode.

	Command	Purpose
Step 2	**activation-key** *key* **Example:** `active(config)# activation-key 0xd11b3d48 0xa80a4c0a 0x48e0fd1c 0xb0443480 0x843fc490`	Installs the new license on the active unit.
	On the standby unit:	
Step 3	**activation-key** *key* **Example:** `standby(config)# activation-key 0xc125727f 0x903de1ee 0x8c838928 0x92dc84d4 0x003a2ba0`	Installs the new license on the standby unit.
	On the active unit:	
Step 4	**failover** **Example:** `active(config)# failover`	Reenables failover.

Upgrading the License for a Failover (Reload Required)

Use the following procedure if your new license requires you to reload. See Table 3-9 on page 3-15 for more information about licenses requiring a reload. Reloading the failover pair causes a loss of connectivity during the reload.

Detailed Steps

	Command	Purpose
	On the active unit:	
Step 1	**no failover** **Example:** `active(config)# no failover`	Disables failover on the active unit. The standby unit remains in standby mode.
Step 2	**activation-key** *key* **Example:** `active(config)# activation-key 0xd11b3d48 0xa80a4c0a 0x48e0fd1c 0xb0443480 0x843fc490`	Installs the new license on the active unit. If you need to reload, you will see the following message: `WARNING: The running activation key was not updated with the requested key. The flash activation key was updated with the requested key, and will become active after the next reload.` If you do not need to reload, then follow the "Upgrading the License for a Failover (No Reload Required)" section on page 3-16 instead of this procedure.
	On the standby unit:	

	Command	Purpose
Step 3	**activation-key** *key* **Example:** `standby(config)# activation-key 0xc125727f 0x903de1ee 0x8c838928 0x92dc84d4 0x003a2ba0`	Installs the new license on the standby unit.
Step 4	**reload** **Example:** `standby(config)# reload`	Reloads the standby unit.
	On the active unit:	
Step 5	**reload** **Example:** `active(config)# reload`	Reloads the active unit. When you are prompted to save the configuration before reloading, answer **No**. This means that when the active unit comes back up, failover will still be enabled.

Feature History for Licensing

Table 3-10 lists the release history for this feature.

Table 3-10 Feature History for Licensing

Feature Name	Releases	Feature Information
Increased Connections and VLANs	7.0(5)	Increased the following limits: • ASA5510 Base license connections from 32000 to 5000; VLANs from 0 to 10. • ASA5510 Security Plus license connections from 64000 to 130000; VLANs from 10 to 25. • ASA5520 connections from 130000 to 280000; VLANs from 25 to 100. • ASA5540 connections from 280000 to 400000; VLANs from 100 to 200.
SSL VPN Licenses for the ASA 5500 series	7.1(1)	SSL VPN licenses were introduced. This feature is not supported on the Cisco PIX 500 series.
Increased SSL VPN Licenses	7.2(1)	A 5000-user SSL VPN license was introduced for the ASA 5550 and above.

Table 3-10 Feature History for Licensing (continued)

Feature Name	Releases	Feature Information
Increased VLANs	7.2(2)	The maximum number of VLANs for the Security Plus License on the ASA 5505 was increased from 5 (3 fully functional; 1 failover; one restricted to a backup interface) to 20 fully functional interfaces. In addition, the number of trunk ports was increased from 1 to 8. Now there are 20 fully functional interfaces, you do not need to use the **backup interface** command to cripple a backup ISP interface; you can use a fully-functional interface for it. The **backup interface** command is still useful for an Easy VPN configuration. VLAN limits were also increased for the ASA 5510 (from 10 to 50 for the Base License, and from 25 to 100 for the Security Plus License), the ASA 5520 (from 100 to 150), the ASA 5550 (from 200 to 250).
Gigabit Ethernet Support for the ASA 5510	7.2(3)	The ASA 5510 now has the Security Plus License to enable GE (Gigabit Ethernet) for port 0 and 1. If you upgrade the license from Base to Security Plus, the capacity of the external Ethernet0/0 and Ethernet0/1 ports increases from the original FE (Fast Ethernet) (100 Mbps) to GE (1000 Mbps). The interface names will remain Ethernet 0/0 and Ethernet 0/1. Use the **speed** command to change the speed on the interface and use the **show interface** command to see what speed is currently configured for each interface.
Advanced Endpoint Assessment License for the ASA 5500 series	8.0(2)	The Advanced Endpoint Assessment License was introduced. As a condition for the completion of a Cisco AnyConnect or clientless SSL VPN connections, the remote computer scans for a greatly expanded collection of antivirus and antispyware applications, firewalls, operating systems, and associated updates. It also scans for any registry entries, filenames, and process names that you specify. It sends the scan results to the adaptive security appliance. The security appliance uses both the user login credentials and the computer scan results to assign a Dynamic Access Policy (DAP). With an Advanced Endpoint Assessment License, you can enhance Host Scan by configuring an attempt to update noncompliant computers to meet version requirements. We provide timely updates to the list of applications and versions that Host Scan supports in a package that is separate from Cisco Secure Desktop. This feature is not supported on the PIX 500 series.
VPN Load Balancing for the ASA 5510	8.0(2)	VPN load balancing is now supported on the ASA 5510 Security Plus License.

Table 3-10 Feature History for Licensing (continued)

Feature Name	Releases	Feature Information
VPN Flex and Temporary Licenses for the ASA 5500 series	8.0(4)	Support for temporary licenses was introduced. This feature is not supported on the PIX 500 series.
Unified Communications Proxy Sessions license for the ASA 5500 series	8.0(4)	The UC Proxy sessions license was introduced. This feature is not available in Version 8.1. This feature is not supported on the PIX 500 series.



CHAPTER 4

Enabling Multiple Context Mode

This chapter describes how to use security contexts and enable multiple context mode. This chapter includes the following sections:

- Security Context Overview, page 4-1
- Enabling or Disabling Multiple Context Mode, page 4-10

Security Context Overview

You can partition a single security appliance into multiple virtual devices, known as security contexts. Each context is an independent device, with its own security policy, interfaces, and administrators. Multiple contexts are similar to having multiple standalone devices. Many features are supported in multiple context mode, including routing tables, firewall features, IPS, and management. Some features are not supported, including VPN and dynamic routing protocols.

Note When the security appliance is configured for security contexts (also called firewall multmode) or Active/Active stateful failover, IPSec or SSL VPN cannot be enabled. Therefore, these features are unavailable.

This section provides an overview of security contexts, and includes the following topics:

- Common Uses for Security Contexts, page 4-2
- Unsupported Features, page 4-2
- Context Configuration Files, page 4-2
- How the Security Appliance Classifies Packets, page 4-3
- Cascading Security Contexts, page 4-8
- Management Access to Security Contexts, page 4-9

Common Uses for Security Contexts

You might want to use multiple security contexts in the following situations:

- You are a service provider and want to sell security services to many customers. By enabling multiple security contexts on the security appliance, you can implement a cost-effective, space-saving solution that keeps all customer traffic separate and secure, and also eases configuration.
- You are a large enterprise or a college campus and want to keep departments completely separate.
- You are an enterprise that wants to provide distinct security policies to different departments.
- You have any network that requires more than one security appliance.

Unsupported Features

Multiple context mode does not support the following features:

- Dynamic routing protocols

  Security contexts support only static routes. You cannot enable OSPF, RIP, or EIGRP in multiple context mode.
- VPN
- Multicast routing. Multicast bridging is supported.
- Threat Detection
- QoS
- Phone Proxy

Context Configuration Files

This section describes how the security appliance implements multiple context mode configurations and includes the following sections:

- Context Configurations, page 4-2
- System Configuration, page 4-3
- Admin Context Configuration, page 4-3

Context Configurations

The security appliance includes a configuration for each context that identifies the security policy, interfaces, and almost all the options you can configure on a standalone device. You can store context configurations on the internal Flash memory or the external Flash memory card, or you can download them from a TFTP, FTP, or HTTP(S) server.

System Configuration

The system administrator adds and manages contexts by configuring each context configuration location, allocated interfaces, and other context operating parameters in the system configuration, which, like a single mode configuration, is the startup configuration. The system configuration identifies basic settings for the security appliance. The system configuration does not include any network interfaces or network settings for itself; rather, when the system needs to access network resources (such as downloading the contexts from the server), it uses one of the contexts that is designated as the *admin context*. The system configuration does include a specialized failover interface for failover traffic only.

Admin Context Configuration

The admin context is just like any other context, except that when a user logs in to the admin context, then that user has system administrator rights and can access the system and all other contexts. The admin context is not restricted in any way, and can be used as a regular context. However, because logging into the admin context grants you administrator privileges over all contexts, you might need to restrict access to the admin context to appropriate users. The admin context must reside on Flash memory, and not remotely.

If your system is already in multiple context mode, or if you convert from single mode, the admin context is created automatically as a file on the internal Flash memory called admin.cfg. This context is named "admin." If you do not want to use admin.cfg as the admin context, you can change the admin context.

How the Security Appliance Classifies Packets

Each packet that enters the security appliance must be classified, so that the security appliance can determine to which context to send a packet. This section includes the following topics:

- Valid Classifier Criteria, page 4-3
- Invalid Classifier Criteria, page 4-4
- Classification Examples, page 4-5

Note If the destination MAC address is a multicast or broadcast MAC address, the packet is duplicated and delivered to each context.

Valid Classifier Criteria

This section describes the criteria used by the classifier, and includes the following topics:

- Unique Interfaces, page 4-3
- Unique MAC Addresses, page 4-4
- NAT Configuration, page 4-4

Unique Interfaces

If only one context is associated with the ingress interface, the security appliance classifies the packet into that context. In transparent firewall mode, unique interfaces for contexts are required, so this method is used to classify packets at all times.

Unique MAC Addresses

If multiple contexts share an interface, then the classifier uses the interface MAC address. The security appliance lets you assign a different MAC address in each context to the same shared interface, whether it is a shared physical interface or a shared subinterface. By default, shared interfaces do not have unique MAC addresses; the interface uses the physical interface burned-in MAC address in every context. An upstream router cannot route directly to a context without unique MAC addresses. You can set the MAC addresses manually when you configure each interface (see the "Configuring Interface Parameters" section on page 8-2), or you can automatically generate MAC addresses (see the "Automatically Assigning MAC Addresses to Context Interfaces" section on page 7-11).

NAT Configuration

If you do not have unique MAC addresses, then the classifier intercepts the packet and performs a destination IP address lookup. All other fields are ignored; only the destination IP address is used. To use the destination address for classification, the classifier must have knowledge about the subnets located behind each security context. The classifier relies on the NAT configuration to determine the subnets in each context. The classifier matches the destination IP address to either a **static** command or a **global** command. In the case of the **global** command, the classifier does not need a matching **nat** command or an active NAT session to classify the packet. Whether the packet can communicate with the destination IP address after classification depends on how you configure NAT and NAT control.

For example, the classifier gains knowledge about subnets 10.10.10.0, 10.20.10.0 and 10.30.10.0 when the context administrators configure **static** commands in each context:

- Context A:

```
static (inside,shared) 10.10.10.0 10.10.10.0 netmask 255.255.255.0
```

- Context B:

```
static (inside,shared) 10.20.10.0 10.20.10.0 netmask 255.255.255.0
```

- Context C:

```
static (inside,shared) 10.30.10.0 10.30.10.0 netmask 255.255.255.0
```

Note For management traffic destined for an interface, the interface IP address is used for classification.

Invalid Classifier Criteria

The following configurations are not used for packet classification:

- NAT exemption—The classifier does not use a NAT exemption configuration for classification purposes because NAT exemption does not identify a mapped interface.
- Routing table—If a context includes a static route that points to an external router as the next-hop to a subnet, and a different context includes a **static** command for the same subnet, then the classifier uses the **static** command to classify packets destined for that subnet and ignores the static route.

Classification Examples

Figure 4-1 shows multiple contexts sharing an outside interface. The classifier assigns the packet to Context B because Context B includes the MAC address to which the router sends the packet.

Figure 4-1 Packet Classification with a Shared Interface using MAC Addresses

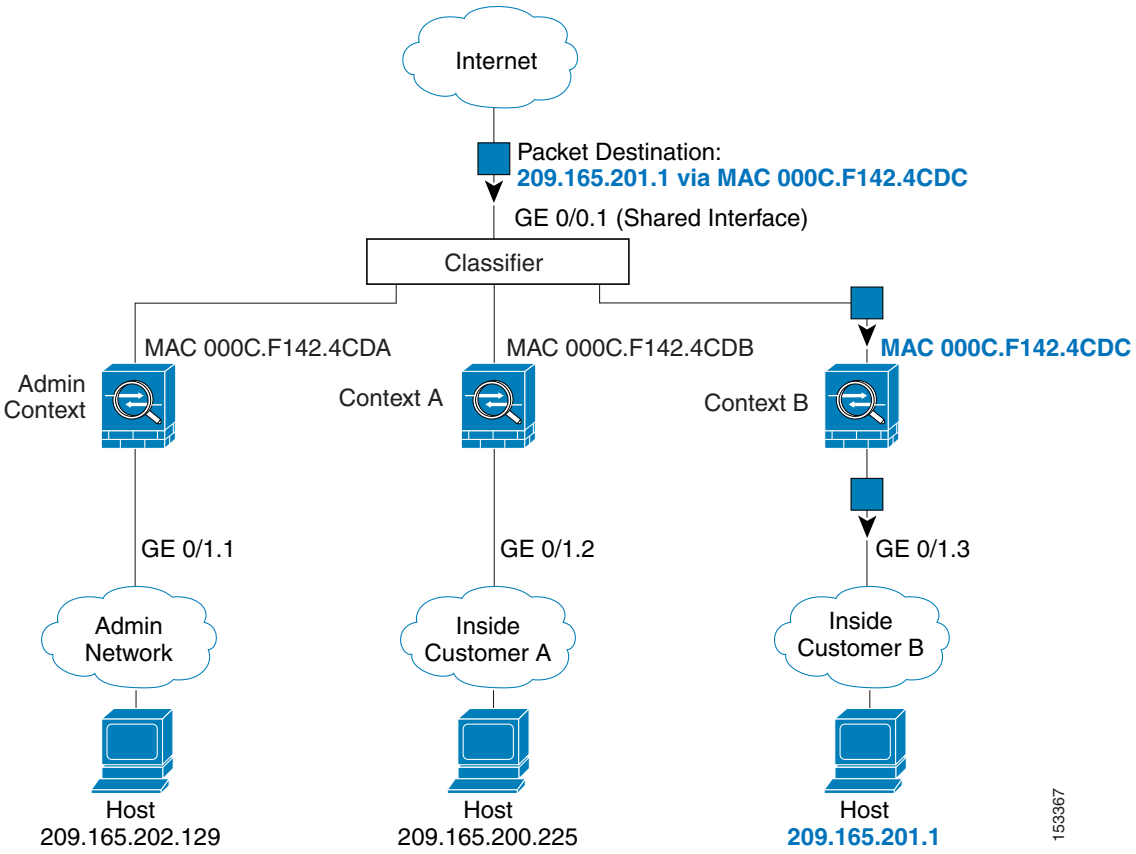
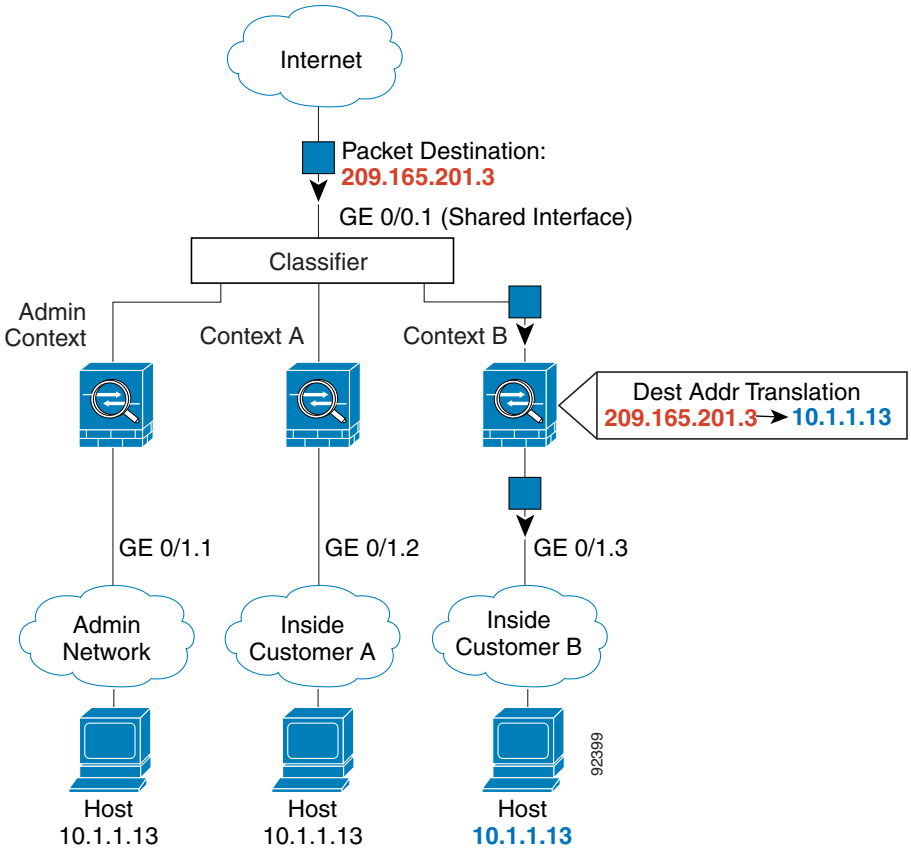


Figure 4-2 shows multiple contexts sharing an outside interface without MAC addresses assigned. The classifier assigns the packet to Context B because Context B includes the address translation that matches the destination address.

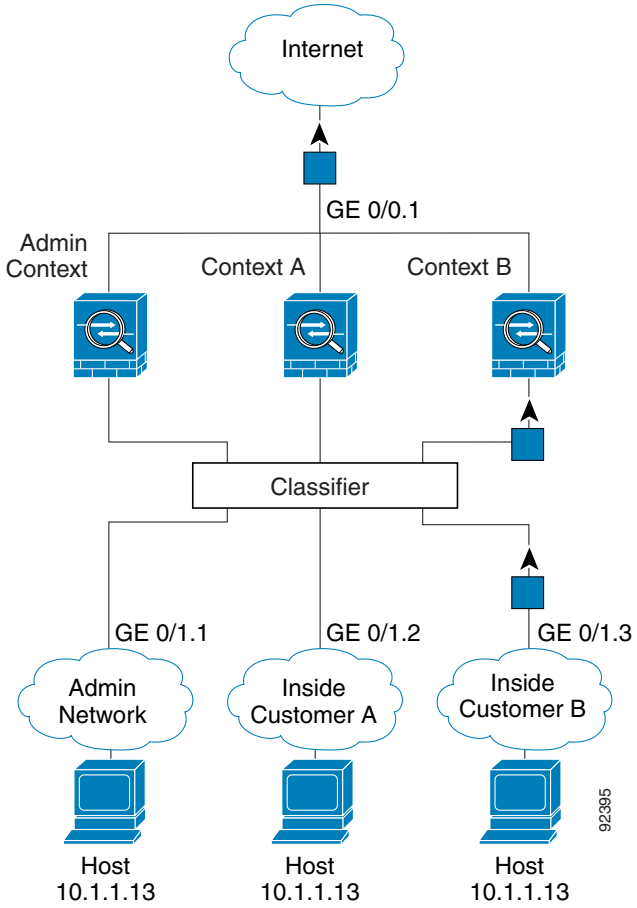
Figure 4-2 Packet Classification with a Shared Interface using NAT



Note that all new incoming traffic must be classified, even from inside networks. Figure 4-3 shows a host on the Context B inside network accessing the Internet. The classifier assigns the packet to Context B because the ingress interface is Gigabit Ethernet 0/1.3, which is assigned to Context B.

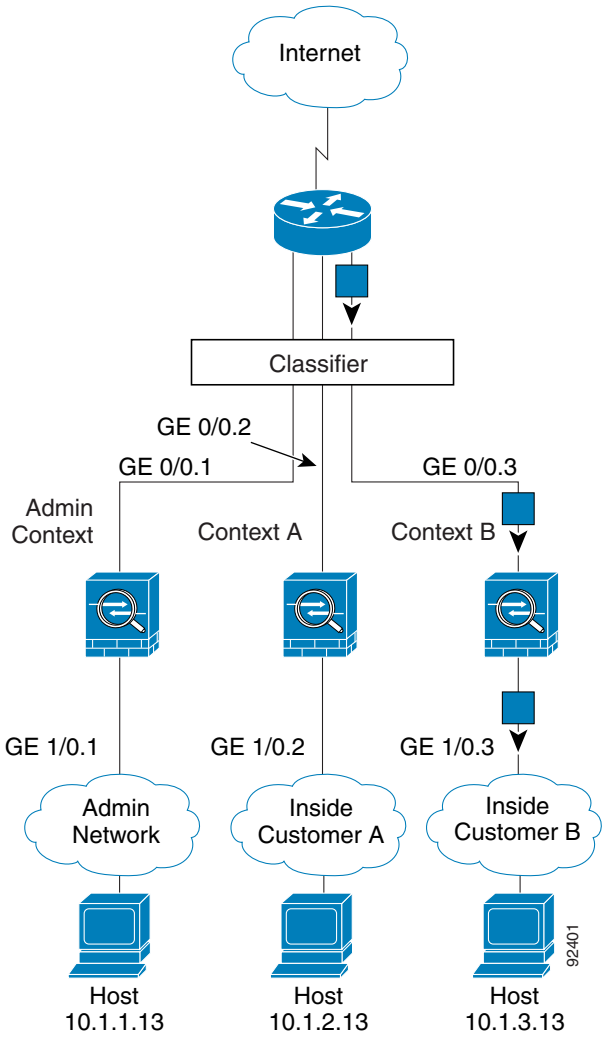
Note If you share an *inside* interface and do not use unique MAC addresses, the classifier imposes some major restrictions. The classifier relies on the address translation configuration to classify the packet within a context, and you must translate the *destination* addresses of the traffic. Because you do not usually perform NAT on outside addresses, sending packets from inside to outside on a shared interface is not always possible; the outside network is large, (the Web, for example), and addresses are not predictable for an outside NAT configuration. If you share an inside interface, we suggest you use unique MAC addresses.

Figure 4-3 Incoming Traffic from Inside Networks



For transparent firewalls, you must use unique interfaces. Figure 4-4 shows a host on the Context B inside network accessing the Internet. The classifier assigns the packet to Context B because the ingress interface is Gigabit Ethernet 1/0.3, which is assigned to Context B.

Figure 4-4 Transparent Firewall Contexts



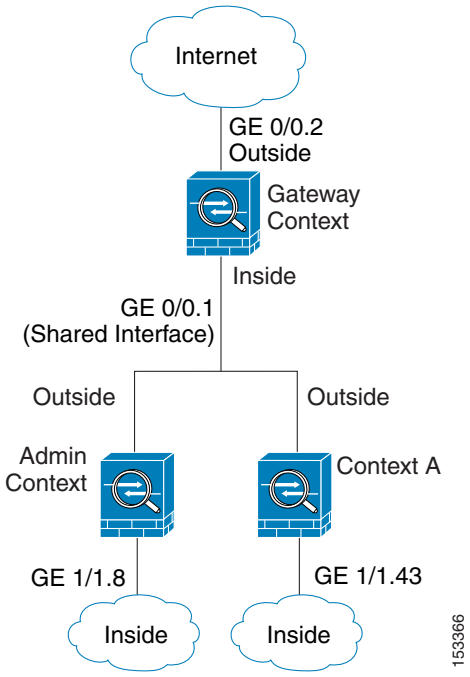
Cascading Security Contexts

Placing a context directly in front of another context is called cascading contexts; the outside interface of one context is the same interface as the inside interface of another context. You might want to cascade contexts if you want to simplify the configuration of some contexts by configuring shared parameters in the top context.

Note Cascading contexts requires that you configure unique MAC addresses for each context interface. Because of the limitations of classifying packets on shared interfaces without MAC addresses, we do not recommend using cascading contexts without unique MAC addresses.

Figure 4-5 shows a gateway context with two contexts behind the gateway.

Figure 4-5 Cascading Contexts



Management Access to Security Contexts

The security appliance provides system administrator access in multiple context mode as well as access for individual context administrators. The following sections describe logging in as a system administrator or as a a context administrator:

- System Administrator Access, page 4-9
- Context Administrator Access, page 4-10

System Administrator Access

You can access the security appliance as a system administrator in two ways:

- Access the security appliance console.

  From the console, you access the *system execution space*, which means that any commands you enter affect only the system configuration or the running of the system (for run-time commands).

- Access the admin context using Telnet, SSH, or ASDM.

  See Chapter 42, "Managing System Access," to enable Telnet, SSH, and SDM access.

As the system administrator, you can access all contexts.

When you change to a context from admin or the system, your username changes to the default "enable_15" username. If you configured command authorization in that context, you need to either configure authorization privileges for the "enable_15" user, or you can log in as a different name for which you provide sufficient privileges in the command authorization configuration for the context. To

log in with a username, enter the **login** command. For example, you log in to the admin context with the username "admin." The admin context does not have any command authorization configuration, but all other contexts include command authorization. For convenience, each context configuration includes a user "admin" with maximum privileges. When you change from the admin context to context A, your username is altered, so you must log in again as "admin" by entering the **login** command. When you change to context B, you must again enter the **login** command to log in as "admin."

The system execution space does not support any AAA commands, but you can configure its own enable password, as well as usernames in the local database to provide individual logins.

Context Administrator Access

You can access a context using Telnet, SSH, or ASDM. If you log in to a non-admin context, you can only access the configuration for that context. You can provide individual logins to the context. See See Chapter 42, "Managing System Access," to enable Telnet, SSH, and SDM access and to configure management authentication.

Enabling or Disabling Multiple Context Mode

Your security appliance might already be configured for multiple security contexts depending on how you ordered it from Cisco. If you are upgrading, however, you might need to convert from single mode to multiple mode by following the procedures in this section. ASDM does not support changing modes, so you need to change modes using the CLI.

This section includes the following topics:

- Backing Up the Single Mode Configuration, page 4-10
- Enabling Multiple Context Mode, page 4-10
- Restoring Single Context Mode, page 4-11

Backing Up the Single Mode Configuration

When you convert from single mode to multiple mode, the security appliance converts the running configuration into two files. The original startup configuration is not saved, so if it differs from the running configuration, you should back it up before proceeding.

Enabling Multiple Context Mode

The context mode (single or multiple) is not stored in the configuration file, even though it does endure reboots. If you need to copy your configuration to another device, set the mode on the new device to match using the **mode** command.

When you convert from single mode to multiple mode, the security appliance converts the running configuration into two files: a new startup configuration that comprises the system configuration, and admin.cfg that comprises the admin context (in the root directory of the internal Flash memory). The original running configuration is saved as old_running.cfg (in the root directory of the internal Flash memory). The original startup configuration is not saved. The security appliance automatically adds an entry for the admin context to the system configuration with the name "admin."

To enable multiple mode, enter the following command:

```
hostname(config)# mode multiple
```

You are prompted to reboot the security appliance.

Restoring Single Context Mode

If you convert from multiple mode to single mode, you might want to first copy a full startup configuration (if available) to the security appliance; the system configuration inherited from multiple mode is not a complete functioning configuration for a single mode device. Because the system configuration does not have any network interfaces as part of its configuration, you must access the security appliance from the console to perform the copy.

To copy the old running configuration to the startup configuration and to change the mode to single mode, perform the following steps in the system execution space:

Step 1 To copy the backup version of your original running configuration to the current startup configuration, enter the following command in the system execution space:

```
hostname(config)# copy flash:old_running.cfg startup-config
```

Step 2 To set the mode to single mode, enter the following command in the system execution space:

```
hostname(config)# mode single
```

The security appliance reboots.



CHAPTER 5

Configuring Switch Ports and VLAN Interfaces for the Cisco ASA 5505 Adaptive Security Appliance

This chapter describes how to configure the switch ports and VLAN interfaces of the ASA 5505 adaptive security appliance.

Note To configure interfaces of other models, see Chapter 6, "Configuring Ethernet Settings, Redundant Interfaces, and Subinterfaces," and Chapter 8, "Configuring Interface Parameters."

The security appliance interfaces do not support jumbo frames.

This chapter includes the following sections:

- Interface Overview, page 5-1
- Configuring VLAN Interfaces, page 5-5
- Configuring Switch Ports as Access Ports, page 5-9
- Configuring a Switch Port as a Trunk Port, page 5-11
- Allowing Communication Between VLAN Interfaces on the Same Security Level, page 5-13

Interface Overview

This section describes the ports and interfaces of the ASA 5505 adaptive security appliance, and includes the following topics:

- Understanding ASA 5505 Ports and Interfaces, page 5-2
- Maximum Active VLAN Interfaces for Your License, page 5-2
- Default Interface Configuration, page 5-4
- VLAN MAC Addresses, page 5-4
- Power Over Ethernet, page 5-4
- Security Level Overview, page 5-5

Understanding ASA 5505 Ports and Interfaces

The ASA 5505 adaptive security appliance supports a built-in switch. There are two kinds of ports and interfaces that you need to configure:

- Physical switch ports—The adaptive security appliance has eight Fast Ethernet switch ports that forward traffic at Layer 2, using the switching function in hardware. Two of these ports are PoE ports. See the "Power Over Ethernet" section on page 5-4 for more information. You can connect these interfaces directly to user equipment such as PCs, IP phones, or a DSL modem. Or you can connect to another switch.
- Logical VLAN interfaces—In routed mode, these interfaces forward traffic between VLAN networks at Layer 3, using the configured security policy to apply firewall and VPN services. In transparent mode, these interfaces forward traffic between the VLANs on the same network at Layer 2, using the configured security policy to apply firewall services. See the "Maximum Active VLAN Interfaces for Your License" section for more information about the maximum VLAN interfaces. VLAN interfaces let you divide your equipment into separate VLANs, for example, home, business, and Internet VLANs.

To segregate the switch ports into separate VLANs, you assign each switch port to a VLAN interface. Switch ports on the same VLAN can communicate with each other using hardware switching. But when a switch port on VLAN 1 wants to communicate with a switch port on VLAN 2, then the adaptive security appliance applies the security policy to the traffic and routes or bridges between the two VLANs.

Note Subinterfaces are not available for the ASA 5505 adaptive security appliance.

Maximum Active VLAN Interfaces for Your License

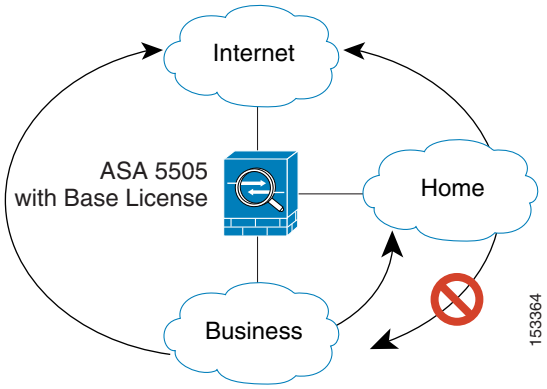
In transparent firewall mode, you can configure two active VLANs in the Base license and three active VLANs in the Security Plus license, one of which must be for failover.

In routed mode, you can configure up to three active VLANs with the Base license, and up to 20 active VLANs with the Security Plus license.

An active VLAN is a VLAN with a **nameif** command configured.

With the Base license, the third VLAN can only be configured to initiate traffic to one other VLAN. See Figure 5-1 for an example network where the Home VLAN can communicate with the Internet, but cannot initiate contact with Business.

Figure 5-1 ASA 5505 Adaptive Security Appliance with Base License

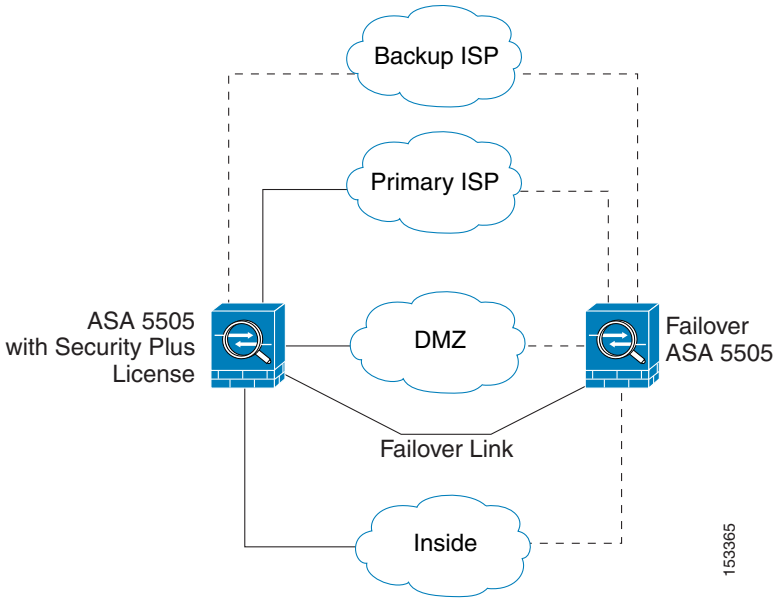


With the Security Plus license, you can configure 20 VLAN interfaces, including a VLAN interface for failover and a VLAN interface as a backup link to your ISP. This backup interface does not pass through traffic unless the route through the primary interface fails. You can configure trunk ports to accomodate multiple VLANs per port.

Note The ASA 5505 adaptive security appliance supports Active/Standby failover, but not Stateful failover.

See Figure 5-2 for an example network.

Figure 5-2 ASA 5505 Adaptive Security Appliance with Security Plus License



Default Interface Configuration

If your adaptive security appliance includes the default factory configuration, your interfaces are configured as follows:

- The outside interface (security level 0) is VLAN 2.

  Ethernet0/0 is assigned to VLAN 2 and is enabled.

  The VLAN 2 IP address is obtained from the DHCP server.

- The inside interface (security level 100) is VLAN 1

  Ethernet 0/1 through Ethernet 0/7 are assigned to VLAN 1 and is enabled.

  VLAN 1 has IP address 192.168.1.1.

Restore the default factory configuration using the **configure factory-default** command.

Use the procedures in this chapter to modify the default configuration, for example, to add VLAN interfaces.

If you do not have a factory default configuration, all switch ports are in VLAN 1, but no other parameters are configured.

VLAN MAC Addresses

In routed firewall mode, all VLAN interfaces share a MAC address. Ensure that any connected switches can support this scenario. If the connected switches require unique MAC addresses, you can manually assign MAC addresses.

In transparent firewall mode, each VLAN has a unique MAC address. You can override the generated MAC addresses if desired by manually assigning MAC addresses.

Power Over Ethernet

Ethernet 0/6 and Ethernet 0/7 support PoE for devices such as IP phones or wireless access points. If you install a non-PoE device or do not connect to these switch ports, the adaptive security appliance does not supply power to the switch ports.

If you shut down the switch port using the **shutdown** command, you disable power to the device. Power is restored when you enter **no shutdown**. See the "Configuring Switch Ports as Access Ports" section on page 5-9 for more information about shutting down a switch port.

To view the status of PoE switch ports, including the type of device connected (Cisco or IEEE 802.3af), use the **show power inline** command.

Monitoring Traffic Using SPAN

If you want to monitor traffic that enters or exits one or more switch ports, you can enable SPAN, also known as switch port monitoring. The port for which you enable SPAN (called the destination port) receives a copy of every packet transmitted or received on a specified source port. The SPAN feature lets you attach a sniffer to the destination port so you can monitor all traffic; without SPAN, you would have to attach a sniffer to every port you want to monitor. You can only enable SPAN for one destination port.

See the **switchport monitor** command in the *Cisco Security Appliance Command Reference* for more information.

Security Level Overview

Each VLAN interface must have a security level in the range 0 to 100 (from lowest to highest). For example, you should assign your most secure network, such as the inside business network, to level 100. The outside network connected to the Internet can be level 0. Other networks, such as a home network can be in between. You can assign interfaces to the same security level. See the "Allowing Communication Between VLAN Interfaces on the Same Security Level" section on page 5-13 for more information.

The level controls the following behavior:

- Network access—By default, there is an implicit permit from a higher security interface to a lower security interface (outbound). Hosts on the higher security interface can access any host on a lower security interface. You can limit access by applying an access list to the interface.

  For same security interfaces, there is an implicit permit for interfaces to access other interfaces on the same security level or lower.

- Inspection engines—Some application inspection engines are dependent on the security level. For same security interfaces, inspection engines apply to traffic in either direction.
  - NetBIOS inspection engine—Applied only for outbound connections.
  - SQL*Net inspection engine—If a control connection for the SQL*Net (formerly OraServ) port exists between a pair of hosts, then only an inbound data connection is permitted through the adaptive security appliance.

- Filtering—HTTP(S) and FTP filtering applies only for outbound connections (from a higher level to a lower level).

  For same security interfaces, you can filter traffic in either direction.

- NAT control—When you enable NAT control, you must configure NAT for hosts on a higher security interface (inside) when they access hosts on a lower security interface (outside).

  Without NAT control, or for same security interfaces, you can choose to use NAT between any interface, or you can choose not to use NAT. Keep in mind that configuring NAT for an outside interface might require a special keyword.

- **established** command—This command allows return connections from a lower security host to a higher security host if there is already an established connection from the higher level host to the lower level host.

  For same security interfaces, you can configure **established** commands for both directions.

Configuring VLAN Interfaces

For each VLAN to pass traffic, you need to configure an interface name (the **nameif** command), and for routed mode, an IP address. You should also change the security level from the default, which is 0. If you name an interface "inside" and you do not set the security level explicitly, then the adaptive security appliance sets the security level to 100.

For information about how many VLANs you can configure, see the "Maximum Active VLAN Interfaces for Your License" section on page 5-2.

Note If you are using failover, do not use this procedure to name interfaces that you are reserving for failover communications. See Chapter 15, "Configuring Failover," to configure the failover link.

If you change the security level of an interface, and you do not want to wait for existing connections to time out before the new security information is used, you can clear the connections using the **clear local-host** command.

To configure a VLAN interface, perform the following steps:

Step 1 To specify the VLAN ID, enter the following command:

```
hostname(config)# interface vlan number
```

Where the *number* is between 1 and 4090.

For example, enter the following command:

```
hostname(config)# interface vlan 100
```

To remove this VLAN interface and all associated configuration, enter the **no interface vlan** command. Because this interface also includes the interface name configuration, and the name is used in other commands, those commands are also removed.

Step 2 (Optional) For the Base license, allow this interface to be the third VLAN by limiting it from initiating contact to one other VLAN using the following command:

```
hostname(config-if)# no forward interface vlan number
```

Where *number* specifies the VLAN ID to which this VLAN interface cannot initiate traffic.

With the Base license, you can only configure a third VLAN if you use this command to limit it.

For example, you have one VLAN assigned to the outside for Internet access, one VLAN assigned to an inside business network, and a third VLAN assigned to your home network. The home network does not need to access the business network, so you can use the **no forward interface** command on the home VLAN; the business network can access the home network, but the home network cannot access the business network.

If you already have two VLAN interfaces configured with a **nameif** command, be sure to enter the **no forward interface** command before the **nameif** command on the third interface; the adaptive security appliance does not allow three fully functioning VLAN interfaces with the Base license on the ASA 5505 adaptive security appliance.

Note If you upgrade to the Security Plus license, you can remove this command and achieve full functionality for this interface. If you leave this command in place, this interface continues to be limited even after upgrading.

Step 3 To name the interface, enter the following command:

```
hostname(config-if)# nameif name
```

The *name* is a text string up to 48 characters, and is not case-sensitive. You can change the name by reentering this command with a new value. Do not enter the **no** form, because that command causes all commands that refer to that name to be deleted.

Step 4 To set the security level, enter the following command:

```
hostname(config-if)# security-level number
```

Where *number* is an integer between 0 (lowest) and 100 (highest).

Step 5 (Routed mode only) To set the IP address, enter one of the following commands.

Note To set an IPv6 address, see the "Configuring IPv6 on an Interface" section on page 13-3.

To set the management IP address for transparent firewall mode, see the "Setting the Management IP Address for a Transparent Firewall" section on page 9-5. In transparent mode, you do not set the IP address for each interface, but rather for the whole adaptive security appliance or context.

For failover, you must set the IP address an standby address manually; DHCP and PPPoE are not supported.

- To set the IP address manually, enter the following command:

```
hostname(config-if)# ip address ip_address [mask] [standby ip_address]
```

The **standby** keyword and address is used for failover. See Chapter 15, "Configuring Failover," for more information.

- To obtain an IP address from a DHCP server, enter the following command:

```
hostname(config-if)# ip address dhcp [setroute]
```

Reenter this command to reset the DHCP lease and request a new lease.

If you do not enable the interface using the **no shutdown** command before you enter the **ip address dhcp** command, some DHCP requests might not be sent.

- To obtain an IP address from a PPPoE server, see Chapter 37, "Configuring the PPPoE Client."

Step 6 (Optional) To assign a private MAC address to this interface, enter the following command:

```
hostname(config-if)# mac-address mac_address [standby mac_address]
```

By default in routed mode, all VLANs use the same MAC address. In transparent mode, the VLANs use unique MAC addresses. You might want to set unique VLANs or change the generated VLANs if your switch requires it, or for access control purposes.

Step 7 (Optional) To set an interface to management-only mode, so that it does not allow through traffic, enter the following command:

```
hostname(config-if)# management-only
```

Step 8 By default, VLAN interfaces are enabled. To enable the interface, if it is not already enabled, enter the following command:

```
hostname(config-if)# no shutdown
```

To disable the interface, enter the **shutdown** command.

The following example configures seven VLAN interfaces, including the failover interface which is configured separately using the **failover lan** command:

```
hostname(config)# interface vlan 100
hostname(config-if)# nameif outside
hostname(config-if)# security-level 0
hostname(config-if)# ip address 10.1.1.1 255.255.255.0
```

```
hostname(config-if)# no shutdown

hostname(config-if)# interface vlan 200
hostname(config-if)# nameif inside
hostname(config-if)# security-level 100
hostname(config-if)# ip address 10.2.1.1 255.255.255.0
hostname(config-if)# no shutdown

hostname(config-if)# interface vlan 201
hostname(config-if)# nameif dept1
hostname(config-if)# security-level 90
hostname(config-if)# ip address 10.2.2.1 255.255.255.0
hostname(config-if)# no shutdown

hostname(config-if)# interface vlan 202
hostname(config-if)# nameif dept2
hostname(config-if)# security-level 90
hostname(config-if)# ip address 10.2.3.1 255.255.255.0
hostname(config-if)# no shutdown

hostname(config-if)# interface vlan 300
hostname(config-if)# nameif dmz
hostname(config-if)# security-level 50
hostname(config-if)# ip address 10.3.1.1 255.255.255.0
hostname(config-if)# no shutdown

hostname(config-if)# interface vlan 400
hostname(config-if)# nameif backup-isp
hostname(config-if)# security-level 50
hostname(config-if)# ip address 10.1.2.1 255.255.255.0
hostname(config-if)# no shutdown

hostname(config-if)# failover lan faillink vlan500
hostname(config)# failover interface ip faillink 10.4.1.1 255.255.255.0 standby 10.4.1.2
255.255.255.0
```

The following example configures three VLAN interfaces for the Base license. The third home interface cannot forward traffic to the business interface.

```
hostname(config)# interface vlan 100
hostname(config-if)# nameif outside
hostname(config-if)# security-level 0
hostname(config-if)# ip address dhcp
hostname(config-if)# no shutdown

hostname(config-if)# interface vlan 200
hostname(config-if)# nameif business
hostname(config-if)# security-level 100
hostname(config-if)# ip address 10.1.1.1 255.255.255.0
hostname(config-if)# no shutdown

hostname(config-if)# interface vlan 300
hostname(config-if)# no forward interface vlan 200
hostname(config-if)# nameif home
hostname(config-if)# security-level 50
hostname(config-if)# ip address 10.2.1.1 255.255.255.0
hostname(config-if)# no shutdown
```

Configuring Switch Ports as Access Ports

By default, all switch ports are shut down. To assign a switch port to one VLAN, configure it as an access port. To create a trunk port to carry multiple VLANs, see the "Configuring a Switch Port as a Trunk Port" section on page 5-11.

By default, the speed and duplex for switch ports are set to auto-negotiate. The default auto-negotiation setting also includes the Auto-MDI/MDIX feature. Auto-MDI/MDIX eliminates the need for crossover cabling by performing an internal crossover when a straight cable is detected during the auto-negotiation phase. Either the speed or duplex must be set to auto-negotiate to enable Auto-MDI/MDIX for the interface. If you explicitly set both the speed and duplex to a fixed value, thus disabling auto-negotiation for both settings, then Auto-MDI/MDIX is also disabled.

Caution The ASA 5505 adaptive security appliance does not support Spanning Tree Protocol for loop detection in the network. Therefore you must ensure that any connection with the adaptive security appliance does not end up in a network loop.

To configure a switch port, perform the following steps:

Step 1 To specify the switch port you want to configure, enter the following command:

```
hostname(config)# interface ethernet0/port
```

Where *port* is 0 through 7. For example, enter the following command:

```
hostname(config)# interface ethernet0/1
```

Step 2 To assign this switch port to a VLAN, enter the following command:

```
hostname(config-if)# switchport access vlan number
```

Where *number* is the VLAN ID, between 1 and 4090.

Note You might assign multiple switch ports to the primary or backup VLANs if the Internet access device includes Layer 2 redundancy.

Step 3 (Optional) To prevent the switch port from communicating with other protected switch ports on the same VLAN, enter the following command:

```
hostname(config-if)# switchport protected
```

You might want to prevent switch ports from communicating with each other if the devices on those switch ports are primarily accessed from other VLANs, you do not need to allow intra-VLAN access, and you want to isolate the devices from each other in case of infection or other security breach. For example, if you have a DMZ that hosts three web servers, you can isolate the web servers from each other if you apply the **switchport protected** command to each switch port. The inside and outside networks can both communicate with all three web servers, and vice versa, but the web servers cannot communicate with each other.

Step 4 (Optional) To set the speed, enter the following command:

```
hostname(config-if)# speed {auto | 10 | 100}
```

The **auto** setting is the default. If you set the speed to anything other than **auto** on PoE ports Ethernet 0/6 or 0/7, then Cisco IP phones and Cisco wireless access points that do not support IEEE 802.3af will not be detected and supplied with power.

Step 5 (Optional) To set the duplex, enter the following command:

```
hostname(config-if)# duplex {auto | full | half}
```

The **auto** setting is the default. If you set the duplex to anything other than **auto** on PoE ports Ethernet 0/6 or 0/7, then Cisco IP phones and Cisco wireless access points that do not support IEEE 802.3af will not be detected and supplied with power.

Step 6 To enable the switch port, if it is not already enabled, enter the following command:

```
hostname(config-if)# no shutdown
```

To disable the switch port, enter the **shutdown** command.

The following example configures five VLAN interfaces, including the failover interface which is configured using the **failover lan** command:

```
hostname(config)# interface vlan 100
hostname(config-if)# nameif outside
hostname(config-if)# security-level 0
hostname(config-if)# ip address 10.1.1.1 255.255.255.0
hostname(config-if)# no shutdown

hostname(config-if)# interface vlan 200
hostname(config-if)# nameif inside
hostname(config-if)# security-level 100
hostname(config-if)# ip address 10.2.1.1 255.255.255.0
hostname(config-if)# no shutdown

hostname(config-if)# interface vlan 300
hostname(config-if)# nameif dmz
hostname(config-if)# security-level 50
hostname(config-if)# ip address 10.3.1.1 255.255.255.0
hostname(config-if)# no shutdown

hostname(config-if)# interface vlan 400
hostname(config-if)# nameif backup-isp
hostname(config-if)# security-level 50
hostname(config-if)# ip address 10.1.2.1 255.255.255.0
hostname(config-if)# no shutdown

hostname(config-if)# failover lan faillink vlan500
hostname(config)# failover interface ip faillink 10.4.1.1 255.255.255.0 standby 10.4.1.2
255.255.255.0

hostname(config)# interface ethernet 0/0
hostname(config-if)# switchport access vlan 100
hostname(config-if)# no shutdown

hostname(config-if)# interface ethernet 0/1
hostname(config-if)# switchport access vlan 200
hostname(config-if)# no shutdown

hostname(config-if)# interface ethernet 0/2
hostname(config-if)# switchport access vlan 300
hostname(config-if)# no shutdown

hostname(config-if)# interface ethernet 0/3
```

```
hostname(config-if)# switchport access vlan 400
hostname(config-if)# no shutdown

hostname(config-if)# interface ethernet 0/4
hostname(config-if)# switchport access vlan 500
hostname(config-if)# no shutdown
```

Configuring a Switch Port as a Trunk Port

By default, all switch ports are shut down. This procedure tells how to create a trunk port that can carry multiple VLANs using 802.1Q tagging. Trunk mode is available only with the Security Plus license.

To create an access port, where an interface is assigned to only one VLAN, see the "Configuring Switch Ports as Access Ports" section on page 5-9.

By default, the speed and duplex for switch ports are set to auto-negotiate. The default auto-negotiation setting also includes the Auto-MDI/MDIX feature. Auto-MDI/MDIX eliminates the need for crossover cabling by performing an internal crossover when a straight cable is detected during the auto-negotiation phase. Either the speed or duplex must be set to auto-negotiate to enable Auto-MDI/MDIX for the interface. If you explicitly set both the speed and duplex to a fixed value, thus disabling auto-negotiation for both settings, then Auto-MDI/MDIX is also disabled.

To configure a trunk port, perform the following steps:

Step 1 To specify the switch port you want to configure, enter the following command:

```
hostname(config)# interface ethernet0/port
```

Where *port* is 0 through 7. For example, enter the following command:

```
hostname(config)# interface ethernet0/1
```

Step 2 To assign VLANs to this trunk, enter one or more of the following commands.

- To assign native VLANs, enter the following command:

  ```
  hostname(config-if)# switchport trunk native vlan vlan_id
  ```

  where the *vlan_id* is a single VLAN ID between 1 and 4090.

  Packets on the native VLAN are not modified when sent over the trunk. For example, if a port has VLANs 2, 3 and 4 assigned to it, and VLAN 2 is the native VLAN, then packets on VLAN 2 that egress the port are not modified with an 802.1Q header. Frames which ingress (enter) this port and have no 802.1Q header are put into VLAN 2.

  Each port can only have one native VLAN, but every port can have either the same or a different native VLAN.

- To assign VLANs, enter the following command:

  ```
  hostname(config-if)# switchport trunk allowed vlan vlan_range
  ```

  where the *vlan_range* (with VLANs between 1 and 4090) can be identified in one of the following ways:

  A single number (n)

  A range (n-x)

  Separate numbers and ranges by commas, for example:

5,7-10,13,45-100

You can enter spaces instead of commas, but the command is saved to the configuration with commas.

You can include the native VLAN in this command, but it is not required; the native VLAN is passed whether it is included in this command or not.

This switch port cannot pass traffic until you assign at least one VLAN to it, native or non-native.

Step 3 To make this switch port a trunk port, enter the following command:

```
hostname(config-if)# switchport mode trunk
```

To restore this port to access mode, enter the **switchport mode access** command.

Step 4 (Optional) To prevent the switch port from communicating with other protected switch ports on the same VLAN, enter the following command:

```
hostname(config-if)# switchport protected
```

You might want to prevent switch ports from communicating with each other if the devices on those switch ports are primarily accessed from other VLANs, you do not need to allow intra-VLAN access, and you want to isolate the devices from each other in case of infection or other security breach. For example, if you have a DMZ that hosts three web servers, you can isolate the web servers from each other if you apply the **switchport protected** command to each switch port. The inside and outside networks can both communicate with all three web servers, and vice versa, but the web servers cannot communicate with each other.

Step 5 (Optional) To set the speed, enter the following command:

```
hostname(config-if)# speed {auto | 10 | 100}
```

The **auto** setting is the default.

Step 6 (Optional) To set the duplex, enter the following command:

```
hostname(config-if)# duplex {auto | full | half}
```

The **auto** setting is the default.

Step 7 To enable the switch port, if it is not already enabled, enter the following command:

```
hostname(config-if)# no shutdown
```

To disable the switch port, enter the **shutdown** command.

The following example configures seven VLAN interfaces, including the failover interface which is configured using the **failover lan** command. VLANs 200, 201, and 202 are trunked on Ethernet 0/1.

```
hostname(config)# interface vlan 100
hostname(config-if)# nameif outside
hostname(config-if)# security-level 0
hostname(config-if)# ip address 10.1.1.1 255.255.255.0
hostname(config-if)# no shutdown

hostname(config-if)# interface vlan 200
hostname(config-if)# nameif inside
hostname(config-if)# security-level 100
hostname(config-if)# ip address 10.2.1.1 255.255.255.0
hostname(config-if)# no shutdown

hostname(config-if)# interface vlan 201
hostname(config-if)# nameif dept1
```

```
hostname(config-if)# security-level 90
hostname(config-if)# ip address 10.2.2.1 255.255.255.0
hostname(config-if)# no shutdown

hostname(config-if)# interface vlan 202
hostname(config-if)# nameif dept2
hostname(config-if)# security-level 90
hostname(config-if)# ip address 10.2.3.1 255.255.255.0
hostname(config-if)# no shutdown

hostname(config-if)# interface vlan 300
hostname(config-if)# nameif dmz
hostname(config-if)# security-level 50
hostname(config-if)# ip address 10.3.1.1 255.255.255.0
hostname(config-if)# no shutdown

hostname(config-if)# interface vlan 400
hostname(config-if)# nameif backup-isp
hostname(config-if)# security-level 50
hostname(config-if)# ip address 10.1.2.1 255.255.255.0
hostname(config-if)# no shutdown

hostname(config-if)# failover lan faillink vlan500
hostname(config)# failover interface ip faillink 10.4.1.1 255.255.255.0 standby 10.4.1.2
255.255.255.0

hostname(config)# interface ethernet 0/0
hostname(config-if)# switchport access vlan 100
hostname(config-if)# no shutdown

hostname(config-if)# interface ethernet 0/1
hostname(config-if)# switchport mode trunk
hostname(config-if)# switchport trunk allowed vlan 200-202
hostname(config-if)# switchport trunk native vlan 5
hostname(config-if)# no shutdown

hostname(config-if)# interface ethernet 0/2
hostname(config-if)# switchport access vlan 300
hostname(config-if)# no shutdown

hostname(config-if)# interface ethernet 0/3
hostname(config-if)# switchport access vlan 400
hostname(config-if)# no shutdown

hostname(config-if)# interface ethernet 0/4
hostname(config-if)# switchport access vlan 500
hostname(config-if)# no shutdown
```

Allowing Communication Between VLAN Interfaces on the Same Security Level

By default, interfaces on the same security level cannot communicate with each other. Allowing communication between same security interfaces lets traffic flow freely between all same security interfaces without access lists.

Note If you enable NAT control, you do not need to configure NAT between same security level interfaces. See the "NAT and Same Security Level Interfaces" section on page 19-15 for more information on NAT and same security level interfaces.

If you enable same security interface communication, you can still configure interfaces at different security levels as usual.

To enable interfaces on the same security level so that they can communicate with each other, enter the following command:

```
hostname(config)# same-security-traffic permit inter-interface
```

To disable this setting, use the **no** form of this command.



CHAPTER 6

Configuring Ethernet Settings, Redundant Interfaces, and Subinterfaces

This chapter describes how to configure and enable physical Ethernet interfaces, how to create redundant interface pairs, and how to add subinterfaces. If you have both fiber and copper Ethernet ports (for example, on the 4GE SSM for the ASA 5510 and higher series adaptive security appliance), this chapter describes how to configure the interface media type.

- In single context mode, complete the procedures in this chapter and then continue your interface configuration in Chapter 8, "Configuring Interface Parameters."
- In multiple context mode, complete the procedures in this chapter in the system execution space, then assign interfaces and subinterfaces to contexts according to Chapter 7, "Adding and Managing Security Contexts," and finally configure the interface parameters within each context according to Chapter 8, "Configuring Interface Parameters."

Note To configure interfaces for the ASA 5505 adaptive security appliance, see Chapter 5, "Configuring Switch Ports and VLAN Interfaces for the Cisco ASA 5505 Adaptive Security Appliance."

The security appliance interfaces do not support jumbo frames.

This chapter includes the following sections:

- Configuring and Enabling RJ-45 Interfaces, page 6-1
- Configuring and Enabling Fiber Interfaces, page 6-3
- Configuring a Redundant Interface, page 6-4
- Configuring VLAN Subinterfaces and 802.1Q Trunking, page 6-7

Configuring and Enabling RJ-45 Interfaces

This section describes how to configure Ethernet settings for physical interfaces with an RJ-45 connector, and how to enable the interface. It includes the following topics:

- RJ-45 Interface Overview, page 6-2
- Configuring the RJ-45 Interface, page 6-2

RJ-45 Interface Overview

This section describes the RJ-45 interface, and includes the following topics:

- Default State of Physical Interfaces, page 6-2
- Connector Types, page 6-2
- Auto-MDI/MDIX Feature, page 6-2

Default State of Physical Interfaces

By default, all physical interfaces are shut down. You must enable the physical interface before any traffic can pass through it (either alone or as part of a redundant interface pair), or through a subinterface. For multiple context mode, if you allocate an interface (physical, redundant, or subinterface) to a context, the interfaces are enabled by default in the context. However, before traffic can pass through the context interface, you must first enable the physical interface in the system configuration according to this procedure.

By default, the speed and duplex for copper (RJ-45) interfaces are set to auto-negotiate.

Connector Types

The ASA 5550 adaptive security appliance and the 4GE SSM for the ASA 5510 and higher adaptive security appliance include two connector types: copper RJ-45 and fiber SFP. RJ-45 is the default. If you want to configure the security appliance to use the fiber SFP connectors, see the "Configuring and Enabling Fiber Interfaces" section on page 6-3.

Auto-MDI/MDIX Feature

For RJ-45 interfaces on the ASA 5500 series adaptive security appliance, the default auto-negotiation setting also includes the Auto-MDI/MDIX feature. Auto-MDI/MDIX eliminates the need for crossover cabling by performing an internal crossover when a straight cable is detected during the auto-negotiation phase. Either the speed or duplex must be set to auto-negotiate to enable Auto-MDI/MDIX for the interface. If you explicitly set both the speed and duplex to a fixed value, thus disabling auto-negotiation for both settings, then Auto-MDI/MDIX is also disabled. For Gigabit Ethernet, when the speed and duplex are set to 1000 and full, then the interface always auto-negotiates; therefore Auto-MDI/MDIX is always enabled and you cannot disable it.

Configuring the RJ-45 Interface

To enable the interface, or to set a specific speed and duplex, perform the following steps:

Step 1 To specify the interface you want to configure, enter the following command:

```
hostname(config)# interface physical_interface
hostname(config-if)#
```

where the *physical_interface* ID includes the type, slot, and port number as *type*[*slot*/]*port*.

The physical interface types include the following:

- **ethernet**

- **gigabitethernet**
- **management** (ASA 5500 only)

For the PIX 500 series security appliance, enter the type followed by the port number, for example, **ethernet0**.

For the ASA 5500 series adaptive security appliance, enter the type followed by *slot*/*port*, for example, **gigabitethernet0/1** or **ethernet 0/1**.

The ASA 5500 management interface is a Fast Ethernet interface designed for management traffic only, and is specified as **management0/0**. You can, however, use it for through traffic if desired (see the **management-only** command). In transparent firewall mode, you can use the management interface (for management purposes) in addition to the two interfaces allowed for through traffic. You can also add subinterfaces to the management interface to provide management in each security context for multiple context mode.

Step 2 (Optional) To set the speed, enter the following command:

```
hostname(config-if)# speed {auto | 10 | 100 | 1000 | nonegotiate}
```

The **auto** setting is the default. The **speed nonegotiate** command disables link negotiation.

Step 3 (Optional) To set the duplex, enter the following command:

```
hostname(config-if)# duplex {auto | full | half}
```

The **auto** setting is the default.

Step 4 To enable the interface, enter the following command:

```
hostname(config-if)# no shutdown
```

To disable the interface, enter the **shutdown** command. If you enter the **shutdown** command, you also shut down all subinterfaces. If you shut down an interface in the system execution space, then that interface is shut down in all contexts that share it.

Configuring and Enabling Fiber Interfaces

This section describes how to configure Ethernet settings for physical interfaces, and how to enable the interface. By default, the connectors used on the 4GE SSM or for built-in interfaces in slot 1 on the ASA 5550 adaptive security appliance are the RJ-45 connectors. To use the fiber SFP connectors, you must set the media type to SFP. The fiber interface has a fixed speed and does not support duplex, but you can set the interface to negotiate link parameters (the default) or not to negotiate.

This section includes the following topics:

- Default State of Physical Interfaces, page 6-3
- Configuring the Fiber Interface, page 6-4

Default State of Physical Interfaces

By default, all physical interfaces are shut down. You must enable the physical interface before any traffic can pass through it (either alone or as part of a redundant interface pair), or through a subinterface. For multiple context mode, if you allocate an interface (physical, redundant, or subinterface) to a

context, the interfaces are enabled by default in the context. However, before traffic can pass through the context interface, you must first enable the physical interface in the system configuration according to this procedure.

Configuring the Fiber Interface

To enable the interface, set the media type, or to set negotiation settings, perform the following steps:

Step 1 To specify the interface you want to configure, enter the following command:

```
hostname(config)# interface gigabitethernet 1/port
hostname(config-if)#
```

The fiber interfaces are available in slot 1 only.

Step 2 To set the media type to SFP, enter the following command:

```
hostname(config-if)# media-type sfp
```

To restore the default RJ-45, enter the **media-type rj45** command.

Step 3 (Optional) To disable link negotiation, enter the following command:

```
hostname(config-if)# speed nonegotiate
```

The default is **no speed nonegotiate**, which sets the speed to 1000 Mbps and enables link negotiation for flow-control parameters and remote fault information. The **speed nonegotiate** command disables link negotiation.

Step 4 To enable the interface, enter the following command:

```
hostname(config-if)# no shutdown
```

To disable the interface, enter the **shutdown** command. If you enter the **shutdown** command, you also shut down all subinterfaces. If you shut down an interface in the system execution space, then that interface is shut down in all contexts that share it.

Configuring a Redundant Interface

A logical redundant interface pairs an active and a standby physical interface. When the active interface fails, the standby interface becomes active and starts passing traffic. You can configure a redundant interface to increase the security appliance reliability. This feature is separate from device-level failover, but you can configure redundant interfaces as well as failover if desired. You can configure up to 8 redundant interface pairs.

All security appliance configuration refers to the logical redundant interface instead of the member physical interfaces.

This section describes how to configure redundant interfaces, and includes the following topics:

- Redundant Interface Overview, page 6-5
- Adding a Redundant Interface, page 6-6
- Changing the Active Interface, page 6-7

Redundant Interface Overview

This section includes overview information about redundant interfaces, and includes the following topics:

- Default State of Redundant Interfaces, page 6-5
- Redundant Interfaces and Failover Guidelines, page 6-5
- Redundant Interface MAC Address, page 6-5
- Physical Interface Guidelines, page 6-5

Default State of Redundant Interfaces

When you add a redundant interface, it is enabled by default. However, the member interfaces must also be enabled to pass traffic.

Redundant Interfaces and Failover Guidelines

Follow these guidelines when adding member interfaces:

- If you want to use a redundant interface for the failover or state link, then you must configure the redundant interface as part of the basic configuration on the secondary unit in addition to the primary unit.
- If you use a redundant interface for the failover or state link, you must put a switch or hub between the two units; you cannot connect them directly. Without the switch or hub, you could have the active port on the primary unit connected directly to the standby port on the secondary unit.
- You can monitor redundant interfaces for failover using the **monitor-interface** command; be sure to reference the logical redundant interface name.
- When the active interface fails over to the standby interface, this activity does not cause the redundant interface to appear to be failed when being monitored for device-level failover. Only when both physical interfaces fail does the redundant interface appear to be failed.
- Redundant interface delay values are configurable, but by default the unit will inherit the default delay values based on the physical type of its member interfaces.

Redundant Interface MAC Address

The redundant interface uses the MAC address of the first physical interface that you add. If you change the order of the member interfaces in the configuration, then the MAC address changes to match the MAC address of the interface that is now listed first. Alternatively, you can assign a MAC address to the redundant interface, which is used regardless of the member interface MAC addresses (see the "Configuring Interface Parameters" section on page 8-2 or the "Automatically Assigning MAC Addresses to Context Interfaces" section on page 7-11). When the active interface fails over to the standby, the same MAC address is maintained so that traffic is not disrupted.

Physical Interface Guidelines

Follow these guidelines when adding member interfaces:

- Both member interfaces must be of the same physical type. For example, both must be Ethernet.

- You cannot add a physical interface to the redundant interface if you configured a name for it. You must first remove the name using the **no nameif** command.

Caution If you are using a physical interface already in your configuration, removing the name will clear any configuration that refers to the interface.

- The only configuration available to physical interfaces that are part of a redundant interface pair are physical parameters (set in the "Configuring and Enabling RJ-45 Interfaces" section on page 6-1 or the "Configuring and Enabling Fiber Interfaces" section on page 6-3), the **description** command, and the **shutdown** command. You can also enter run-time commands like **default** and **help**.
- If you shut down the active interface, then the standby interface becomes active.

Adding a Redundant Interface

You can configure up to 8 redundant interface pairs. To configure a redundant interface, perform the following steps:

Step 1 To add the logical redundant interface, enter the following command:

```
hostname(config)# interface redundant number
hostname(config-if)#
```

where the *number* argument is an integer between 1 and 8.

Step 2 To add the first member interface to the redundant interface, enter the following command:

```
hostname(config-if)# member-interface physical_interface
```

See the "Configuring and Enabling RJ-45 Interfaces" section for a description of the physical interface ID.

After you add the interface, any configuration for it (such as an IP address) is removed.

Step 3 To add the second member interface to the redundant interface, enter the following command:

```
hostname(config-if)# member-interface physical_interface
```

Make sure the second interface is the same physical type as the first interface.

To remove a member interface, enter the **no member-interface** *physical_interface* command. You cannot remove both member interfaces from the redundant interface; the redundant interface requires at least one member interface.

Step 4 To enable the interface (if you previously disabled it), enter the following command:

```
hostname(config-if)# no shutdown
```

By default, the interface is enabled. To disable the interface, enter the **shutdown** command. If you enter the **shutdown** command, you also shut down all subinterfaces. If you shut down an interface in the system execution space, then that interface is shut down in all contexts that share it.

The following example creates two redundant interfaces:

```
hostname(config)# interface redundant 1
hostname(config-if)# member-interface gigabitethernet 0/0
hostname(config-if)# member-interface gigabitethernet 0/1
```

```
hostname(config-if)# interface redundant 2
hostname(config-if)# member-interface gigabitethernet 0/2
hostname(config-if)# member-interface gigabitethernet 0/3
```

Changing the Active Interface

By default, the active interface is the first interface listed in the configuration, if it is available. To view which interface is active, enter the following command:

```
hostname# show interface redundantnumber detail | grep Member
```

For example:

```
hostname# show interface redundant1 detail | grep Member
      Members GigabitEthernet0/3(Active), GigabitEthernet0/2
```

To change the active interface, enter the following command:

```
hostname# redundant-interface redundantnumber active-member physical_interface
```

where the **redundant***number* argument is the redundant interface ID, such as **redundant1**.

The *physical_interface* is the member interface ID that you want to be active.

Configuring VLAN Subinterfaces and 802.1Q Trunking

This section describes how to configure a subinterface, and includes the following topics:

- Subinterface Overview, page 6-7
- Adding a Subinterface, page 6-8

Subinterface Overview

Subinterfaces let you divide a physical or redundant interface into multiple logical interfaces that are tagged with different VLAN IDs. An interface with one or more VLAN subinterfaces is automatically configured as an 802.1Q trunk. Because VLANs allow you to keep traffic separate on a given physical interface, you can increase the number of interfaces available to your network without adding additional physical interfaces or security appliances. This feature is particularly useful in multiple context mode so that you can assign unique interfaces to each context.

This section includes the following topics:

- Default State of Subinterfaces, page 6-7
- Maximum Subinterfaces, page 6-8
- Preventing Untagged Packets on the Physical Interface, page 6-8

Default State of Subinterfaces

When you add a subinterface, it is enabled by default. However, the physical or redundant interface must also be enabled to pass traffic (see the "Configuring and Enabling RJ-45 Interfaces" section on page 6-1, the "Configuring and Enabling Fiber Interfaces" section on page 6-3, or the "Configuring a Redundant Interface" section on page 6-4).

Maximum Subinterfaces

To determine how many subinterfaces are allowed for your platform, see the "Supported Feature Licenses Per Model" section on page 3-1.

Preventing Untagged Packets on the Physical Interface

If you use subinterfaces, you typically do not also want the physical interface to pass traffic, because the physical interface passes untagged packets. This property is also true for the active physical interface in a redundant interface pair. Because the physical or redundant interface must be enabled for the subinterface to pass traffic, ensure that the physical or redundant interface does not pass traffic by leaving out the **nameif** command. If you want to let the physical or redundant interface pass untagged packets, you can configure the **nameif** command as usual. See the "Configuring Interface Parameters" section on page 8-1 for more information about completing the interface configuration.

Adding a Subinterface

To add a subinterface and assign a VLAN to it, perform the following steps:

Step 1 To specify the new subinterface, enter the following command:

```
hostname(config)# interface {physical_interface | redundant number}.subinterface
hostname(config-subif)#
```

See the "Configuring and Enabling RJ-45 Interfaces" section for a description of the physical interface ID.

The **redundant** *number* argument is the redundant interface ID, such as **redundant 1**.

The *subinterface* ID is an integer between 1 and 4294967293.

The following command adds a subinterface to a Gigabit Ethernet interface:

```
hostname(config)# interface gigabitethernet 0/1.100
```

The following command adds a subinterface to a redundant interface:

```
hostname(config)# interface redundant 1.100
```

Step 2 To specify the VLAN for the subinterface, enter the following command:

```
hostname(config-subif)# vlan vlan_id
```

The *vlan_id* is an integer between 1 and 4094. Some VLAN IDs might be reserved on connected switches, so check the switch documentation for more information.

You can only assign a single VLAN to a subinterface, and you cannot assign the same VLAN to multiple subinterfaces. You cannot assign a VLAN to the physical interface. Each subinterface must have a VLAN ID before it can pass traffic. To change a VLAN ID, you do not need to remove the old VLAN ID with the **no** option; you can enter the **vlan** command with a different VLAN ID, and the security appliance changes the old ID.

Step 3 To enable the subinterface (if you previously disabled it), enter the following command:

```
hostname(config-subif)# no shutdown
```

By default, the subinterface is enabled. To disable the interface, enter the **shutdown** command. If you shut down an interface in the system execution space, then that interface is shut down in all contexts that share it.



CHAPTER 7

Adding and Managing Security Contexts

This chapter describes how to configure multiple security contexts on the security appliance, and includes the following sections:

- Configuring Resource Management, page 7-1
- Configuring a Security Context, page 7-7
- Automatically Assigning MAC Addresses to Context Interfaces, page 7-11
- Changing Between Contexts and the System Execution Space, page 7-14
- Managing Security Contexts, page 7-15

For information about how contexts work and how to enable multiple context mode, see Chapter 4, "Enabling Multiple Context Mode."

Configuring Resource Management

By default, all security contexts have unlimited access to the resources of the security appliance, except where maximum limits per context are enforced. However, if you find that one or more contexts use too many resources, and they cause other contexts to be denied connections, for example, then you can configure resource management to limit the use of resources per context.

This section includes the following topics:

- Classes and Class Members Overview, page 7-1
- Configuring a Class, page 7-4

Classes and Class Members Overview

The security appliance manages resources by assigning contexts to resource classes. Each context uses the resource limits set by the class. This section includes the following topics:

- Resource Limits, page 7-2
- Default Class, page 7-3
- Class Members, page 7-4

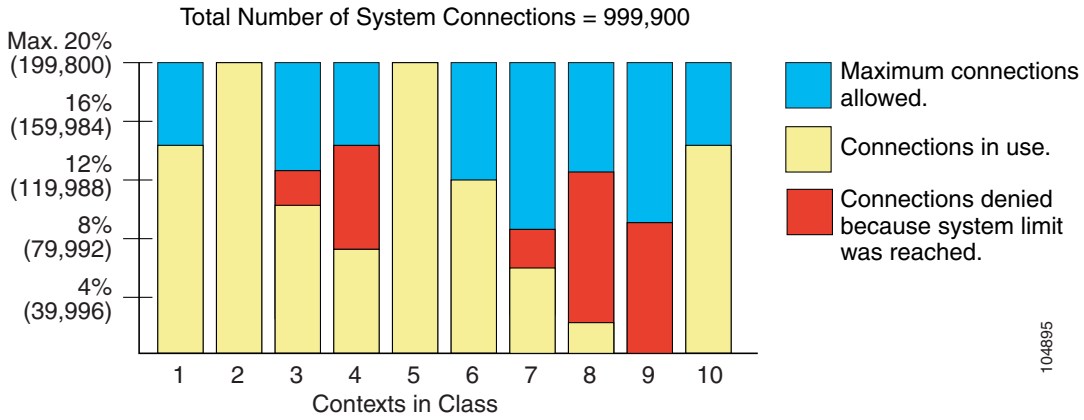
Resource Limits

When you create a class, the security appliance does not set aside a portion of the resources for each context assigned to the class; rather, the security appliance sets the maximum limit for a context. If you oversubscribe resources, or allow some resources to be unlimited, a few contexts can "use up" those resources, potentially affecting service to other contexts.

You can set the limit for individual resources, as a percentage (if there is a hard system limit) or as an absolute value.

You can oversubscribe the security appliance by assigning more than 100 percent of a resource across all contexts. For example, you can set the Bronze class to limit connections to 20 percent per context, and then assign 10 contexts to the class for a total of 200 percent. If contexts concurrently use more than the system limit, then each context gets less than the 20 percent you intended. (See Figure 7-1.)

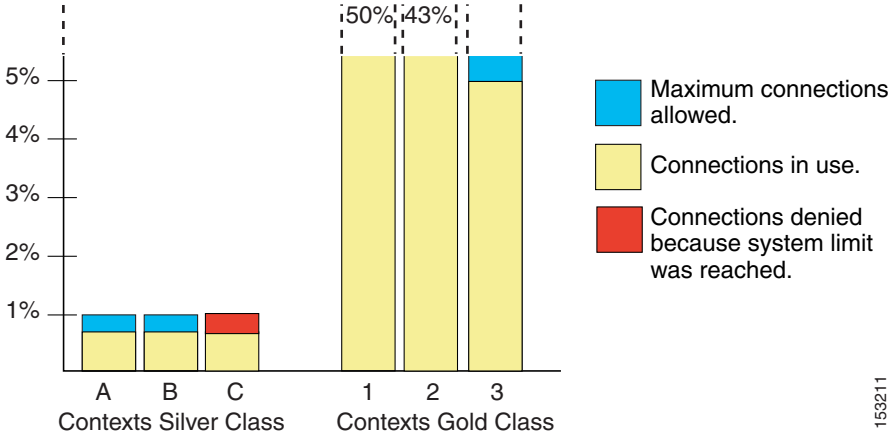
Figure 7-1 Resource Oversubscription



If you assign an absolute value to a resource across all contexts that exceeds the practical limit of the security appliance, then the performance of the security appliance might be impaired.

The security appliance lets you assign unlimited access to one or more resources in a class, instead of a percentage or absolute number. When a resource is unlimited, contexts can use as much of the resource as the system has available or that is practically available. For example, Context A, B, and C are in the Silver Class, which limits each class member to 1 percent of the connections, for a total of 3 percent; but the three contexts are currently only using 2 percent combined. Gold Class has unlimited access to connections. The contexts in the Gold Class can use more than the 97 percent of "unassigned" connections; they can also use the 1 percent of connections not currently in use by Context A, B, and C, even if that means that Context A, B, and C are unable to reach their 3 percent combined limit. (See Figure 7-2.) Setting unlimited access is similar to oversubscribing the security appliance, except that you have less control over how much you oversubscribe the system.

Figure 7-2 Unlimited Resources



Default Class

All contexts belong to the default class if they are not assigned to another class; you do not have to actively assign a context to the default class.

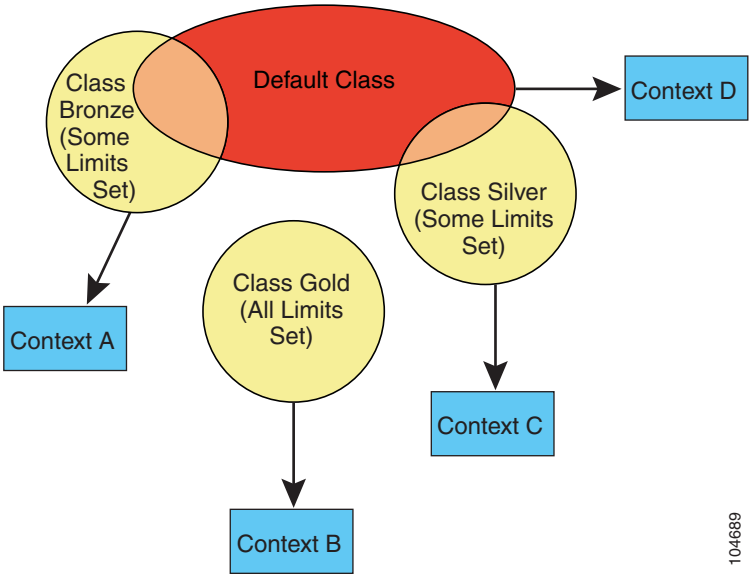
If a context belongs to a class other than the default class, those class settings always override the default class settings. However, if the other class has any settings that are not defined, then the member context uses the default class for those limits. For example, if you create a class with a 2 percent limit for all concurrent connections, but no other limits, then all other limits are inherited from the default class. Conversely, if you create a class with a limit for all resources, the class uses no settings from the default class.

By default, the default class provides unlimited access to resources for all contexts, except for the following limits, which are by default set to the maximum allowed per context:

- Telnet sessions—5 sessions.
- SSH sessions—5 sessions.
- IPSec sessions—5 sessions.
- MAC addresses—65,535 entries.

Figure 7-3 shows the relationship between the default class and other classes. Contexts A and C belong to classes with some limits set; other limits are inherited from the default class. Context B inherits no limits from default because all limits are set in its class, the Gold class. Context D was not assigned to a class, and is by default a member of the default class.

Figure 7-3 Resource Classes



Class Members

To use the settings of a class, assign the context to the class when you define the context. All contexts belong to the default class if they are not assigned to another class; you do not have to actively assign a context to default. You can only assign a context to one resource class. The exception to this rule is that limits that are undefined in the member class are inherited from the default class; so in effect, a context could be a member of default plus another class.

Configuring a Class

To configure a class in the system configuration, perform the following steps. You can change the value of a particular resource limit by reentering the command with a new value.

Step 1 To specify the class name and enter the class configuration mode, enter the following command in the system execution space:

```
hostname(config)# class name
```

The *name* is a string up to 20 characters long. To set the limits for the default class, enter **default** for the name.

Step 2 To set the resource limits, see the following options:

- To set all resource limits (shown in Table 7-1) to be unlimited, enter the following command:

```
hostname(config-resmgmt)# limit-resource all 0
```

For example, you might want to create a class that includes the admin context that has no limitations. The default class has all resources set to unlimited by default.

- To set a particular resource limit, enter the following command:

```
hostname(config-resmgmt)# limit-resource [rate] resource_name number[%]
```

For this particular resource, the limit overrides the limit set for **all**. Enter the **rate** argument to **s**et the rate per second for certain resources. For resources that do not have a system limit, you cannot set the percentage (%) between 1 and 100; you can only set an absolute value. See Table 7-1 for resources for which you can set the rate per second and which to not have a system limit.

Table 7-1 lists the resource types and the limits. See also the **show resource types** command.

Table 7-1 Resource Names and Limits

Resource Name	Rate or Concurrent	Minimum and Maximum Number per Context	System Limit[1]	Description
mac-addresses	Concurrent	N/A	65,535	For transparent firewall mode, the number of MAC addresses allowed in the MAC address table.
conns	Concurrent or Rate	N/A	Concurrent connections: See the "Supported Feature Licenses Per Model" section on page 3-1 for the connection limit for your platform. Rate: N/A	TCP or UDP connections between any two hosts, including connections between one host and multiple other hosts.
inspects	Rate	N/A	N/A	Application inspections.
hosts	Concurrent	N/A	N/A	Hosts that can connect through the security appliance.
asdm	Concurrent	1 minimum 5 maximum	32	ASDM management sessions. **Note** ASDM sessions use two HTTPS connections: one for monitoring that is always present, and one for making configuration changes that is present only when you make changes. For example, the system limit of 32 ASDM sessions represents a limit of 64 HTTPS sessions.
ssh	Concurrent	1 minimum 5 maximum	100	SSH sessions.
syslogs	Rate	N/A	N/A	System log messages.
telnet	Concurrent	1 minimum 5 maximum	100	Telnet sessions.
xlates	Concurrent	N/A	N/A	Address translations.

1. If this column value is N/A, then you cannot set a percentage of the resource because there is no hard system limit for the resource.

For example, to set the default class limit for conns to 10 percent instead of unlimited, enter the following commands:

```
hostname(config)# class default
hostname(config-class)# limit-resource conns 10%
```

All other resources remain at unlimited.

To add a class called gold, enter the following commands:

```
hostname(config)# class gold
```

```
hostname(config-class)# limit-resource mac-addresses 10000
hostname(config-class)# limit-resource conns 15%
hostname(config-class)# limit-resource rate conns 1000
hostname(config-class)# limit-resource rate inspects 500
hostname(config-class)# limit-resource hosts 9000
hostname(config-class)# limit-resource asdm 5
hostname(config-class)# limit-resource ssh 5
hostname(config-class)# limit-resource rate syslogs 5000
hostname(config-class)# limit-resource telnet 5
hostname(config-class)# limit-resource xlates 36000
```

Configuring a Security Context

The security context definition in the system configuration identifies the context name, configuration file URL, and interfaces that a context can use.

Note If you do not have an admin context (for example, if you clear the configuration) then you must first specify the admin context name by entering the following command:

```
hostname(config)# admin-context name
```

Although this context name does not exist yet in your configuration, you can subsequently enter the **context** *name* command to match the specified name to continue the admin context configuration.

To add or change a context in the system configuration, perform the following steps:

Step 1 To add or modify a context, enter the following command in the system execution space:

```
hostname(config)# context name
```

The *name* is a string up to 32 characters long. This name is case sensitive, so you can have two contexts named “customerA” and “CustomerA,” for example. You can use letters, digits, or hyphens, but you cannot start or end the name with a hyphen.

“System” or “Null” (in upper or lower case letters) are reserved names, and cannot be used.

Step 2 (Optional) To add a description for this context, enter the following command:

```
hostname(config-ctx)# description text
```

Step 3 To specify the interfaces you can use in the context, enter the command appropriate for a physical interface or for one or more subinterfaces.

- To allocate a physical interface, enter the following command:

  ```
  hostname(config-ctx)# allocate-interface physical_interface [mapped_name]
  [visible | invisible]
  ```

- To allocate one or more subinterfaces, enter the following command:

  ```
  hostname(config-ctx)# allocate-interface
  physical_interface.subinterface[-physical_interface.subinterface]
  [mapped_name[-mapped_name]] [visible | invisible]
  ```

  Note Do not include a space between the interface type and the port number.

You can enter these commands multiple times to specify different ranges. If you remove an allocation with the **no** form of this command, then any context commands that include this interface are removed from the running configuration.

Transparent firewall mode allows only two interfaces to pass through traffic; however, on the ASA adaptive security appliance, you can use the dedicated management interface, Management 0/0, (either the physical interface or a subinterface) as a third interface for management traffic.

Note The management interface for transparent mode does not flood a packet out the interface when that packet is not in the MAC address table.

You can assign the same interfaces to multiple contexts in routed mode, if desired. Transparent mode does not allow shared interfaces.

The *mapped_name* is an alphanumeric alias for the interface that can be used within the context instead of the interface ID. If you do not specify a mapped name, the interface ID is used within the context. For security purposes, you might not want the context administrator to know which interfaces are being used by the context.

A mapped name must start with a letter, end with a letter or digit, and have as interior characters only letters, digits, or an underscore. For example, you can use the following names:

```
int0

inta

int_0
```

For subinterfaces, you can specify a range of mapped names.

If you specify a range of subinterfaces, you can specify a matching range of mapped names. Follow these guidelines for ranges:

- The mapped name must consist of an alphabetic portion followed by a numeric portion. The alphabetic portion of the mapped name must match for both ends of the range. For example, enter the following range:

  ```
  int0-int10
  ```

  If you enter **gigabitethernet0/1.1-gigabitethernet0/1.5 happy1-sad5**, for example, the command fails.

- The numeric portion of the mapped name must include the same quantity of numbers as the subinterface range. For example, both ranges include 100 interfaces:

  ```
  gigabitethernet0/0.100-gigabitethernet0/0.199 int1-int100
  ```

  If you enter **gigabitethernet0/0.100-gigabitethernet0/0.199 int1-int15**, for example, the command fails.

Specify **visible** to see physical interface properties in the **show interface** command even if you set a mapped name. The default **invisible** keyword specifies to only show the mapped name.

The following example shows gigabitethernet0/1.100, gigabitethernet0/1.200, and gigabitethernet0/2.300 through gigabitethernet0/1.305 assigned to the context. The mapped names are int1 through int8.

```
hostname(config-ctx)# allocate-interface gigabitethernet0/1.100 int1
hostname(config-ctx)# allocate-interface gigabitethernet0/1.200 int2
hostname(config-ctx)# allocate-interface gigabitethernet0/2.300-gigabitethernet0/2.305
int3-int8
```

Step 4 To identify the URL from which the system downloads the context configuration, enter the following command:

```
hostname(config-ctx)# config-url url
```

When you add a context URL, the system immediately loads the context so that it is running, if the configuration is available.

Note Enter the **allocate-interface** command(s) before you enter the **config-url** command. The security appliance must assign interfaces to the context before it loads the context configuration; the context configuration might include commands that refer to interfaces (**interface**, **nat**, **global**...). If you enter the **config-url** command first, the security appliance loads the context configuration immediately. If the context contains any commands that refer to interfaces, those commands fail.

See the following URL syntax:

- **disk:/**[*path*/]*filename*

  This URL indicates the internal Flash memory. The filename does not require a file extension, although we recommend using ".cfg". If the configuration file is not available, you see the following message:

  ```
  WARNING: Could not fetch the URL disk:/url
  INFO: Creating context with default config
  ```

  You can then change to the context, configure it at the CLI, and enter the **write memory** command to write the file to Flash memory.

  Note The admin context file must be stored on the internal Flash memory.

- **ftp://**[*user*[**:***password*]@]*server*[:*port*]**/**[*path*/]*filename*[**;type=***xx*]

  The **type** can be one of the following keywords:

  - **ap**—ASCII passive mode
  - **an**—ASCII normal mode
  - **ip**—(Default) Binary passive mode
  - **in**—Binary normal mode

  The server must be accessible from the admin context. The filename does not require a file extension, although we recommend using ".cfg". If the configuration file is not available, you see the following message:

  ```
  WARNING: Could not fetch the URL ftp://url
  INFO: Creating context with default config
  ```

  You can then change to the context, configure it at the CLI, and enter the **write memory** command to write the file to the FTP server.

- **http**[**s**]**://**[*user*[**:***password*]@]*server*[:*port*]**/**[*path*/]*filename*

  The server must be accessible from the admin context. The filename does not require a file extension, although we recommend using ".cfg". If the configuration file is not available, you see the following message:

  ```
  WARNING: Could not fetch the URL http://url
  INFO: Creating context with default config
  ```

If you change to the context and configure the context at the CLI, you cannot save changes back to HTTP or HTTPS servers using the **write memory** command. You can, however, use the **copy tftp** command to copy the running configuration to a TFTP server.

- **tftp://**[*user*[**:***password*]**@**]*server*[**:***port*]**/**[*path***/**]*filename*[**;int=***interface_name*]

  The server must be accessible from the admin context. Specify the interface name if you want to override the route to the server address. The filename does not require a file extension, although we recommend using ".cfg". If the configuration file is not available, you see the following message:

  ```
  WARNING: Could not fetch the URL tftp://url
  INFO: Creating context with default config
  ```

  You can then change to the context, configure it at the CLI, and enter the **write memory** command to write the file to the TFTP server.

To change the URL, reenter the **config-url** command with a new URL.

See the "Changing the Security Context URL" section on page 7-16 for more information about changing the URL.

For example, enter the following command:

```
hostname(config-ctx)# config-url ftp://joe:passw0rd1@10.1.1.1/configlets/test.cfg
```

Step 5 (Optional) To assign the context to a resource class, enter the following command:

```
hostname(config-ctx)# member class_name
```

If you do not specify a class, the context belongs to the default class. You can only assign a context to one resource class.

For example, to assign the context to the gold class, enter the following command:

```
hostname(config-ctx)# member gold
```

Step 6 (Optional) To assign an IPS virtual sensor to this context if you have the AIP SSM installed, use the **allocate-ips** command. See the "Assigning Virtual Sensors to Security Contexts" section on page 23-6 for detailed information about virtual sensors.

The following example sets the admin context to be "administrator," creates a context called "administrator" on the internal Flash memory, and then adds two contexts from an FTP server:

```
hostname(config)# admin-context administrator
hostname(config)# context administrator
hostname(config-ctx)# allocate-interface gigabitethernet0/0.1
hostname(config-ctx)# allocate-interface gigabitethernet0/1.1
hostname(config-ctx)# config-url flash:/admin.cfg

hostname(config-ctx)# context test
hostname(config-ctx)# allocate-interface gigabitethernet0/0.100 int1
hostname(config-ctx)# allocate-interface gigabitethernet0/0.102 int2
hostname(config-ctx)# allocate-interface gigabitethernet0/0.110-gigabitethernet0/0.115
int3-int8
hostname(config-ctx)# config-url ftp://user1:passw0rd@10.1.1.1/configlets/test.cfg
hostname(config-ctx)# member gold


hostname(config-ctx)# context sample
hostname(config-ctx)# allocate-interface gigabitethernet0/1.200 int1
hostname(config-ctx)# allocate-interface gigabitethernet0/1.212 int2
```

```
hostname(config-ctx)# allocate-interface gigabitethernet0/1.230-gigabitethernet0/1.235
int3-int8
hostname(config-ctx)# config-url ftp://user1:passw0rd@10.1.1.1/configlets/sample.cfg
hostname(config-ctx)# member silver
```

Automatically Assigning MAC Addresses to Context Interfaces

This section tells how to configure auto-generation of MAC addresses, and includes the following sections:

- Information About MAC Addresses, page 7-11
- Default MAC Address, page 7-11
- Failover MAC Addresses, page 7-12
- MAC Address Format, page 7-12
- Enabling Auto-Generation of MAC Addresses, page 7-12
- Viewing Assigned MAC Addresses, page 7-13

Information About MAC Addresses

To allow contexts to share interfaces, we suggest that you assign unique MAC addresses to each shared context interface. The MAC address is used to classify packets within a context. If you share an interface, but do not have unique MAC addresses for the interface in each context, then the destination IP address is used to classify packets. The destination address is matched with the context NAT configuration, and this method has some limitations compared to the MAC address method. See the "How the Security Appliance Classifies Packets" section on page 4-3 for information about classifying packets.

In the rare circumstance that the generated MAC address conflicts with another private MAC address in your network, you can manually set the MAC address for the interface within the context. See the "Configuring the Interface" section on page 8-3 to manually set the MAC address.

Default MAC Address

By default, the physical interface uses the burned-in MAC address, and all subinterfaces of a physical interface use the same burned-in MAC address.

All auto-generated MAC addresses start with A2. The auto-generated MAC addresses are persistent across reloads.

Interaction with Manual MAC Addresses

If you manually assign a MAC address and also enable auto-generation, then the manually assigned MAC address is used. If you later remove the manual MAC address, the auto-generated address is used.

Because auto-generated addresses start with A2, you cannot start manual MAC addresses with A2 if you also want to use auto-generation.

Failover MAC Addresses

For use with failover, the security appliance generates both an active and standby MAC address for each interface. If the active unit fails over and the standby unit becomes active, the new active unit starts using the active MAC addresses to minimize network disruption. See the "MAC Address Format" section for more information.

For upgrading failover units with the legacy version of the **mac-address auto** command before the **prefix** keyword was introduced, see the **mac-address auto** command in the *Cisco Security Appliance Command Reference*.

MAC Address Format

The security appliance generates the MAC address using the following format:

A2xx.yyzz.zzzz

Where xx.yy is a user-defined prefix, and zz.zzzz is an internal counter generated by the security appliance. For the standby MAC address, the address is identical except that the internal counter is increased by 1.

For an example of how the prefix is used, if you set a prefix of 77, then the security appliance converts 77 into the hexadecimal value 004D (yyxx). When used in the MAC address, the prefix is reversed (xxyy) to match the security appliance native form:

A2**4D.00**zz.zzzz

For a prefix of 1009 (03F1), the MAC address is:

A2**F1.03**zz.zzzz

Enabling Auto-Generation of MAC Addresses

You can automatically assign private MAC addresses to each context interface by entering the following command in the system configuration:

```
hostname(config)# mac-address auto prefix prefix
```

Where the *prefix* is a decimal value between 0 and 65535. This prefix is converted to a 4-digit hexadecimal number, and used as part of the MAC address. The prefix ensures that each security appliance uses unique MAC addresses, so you can have multiple security appliances on a network segment, for example. See the "MAC Address Format" section for more information about how the prefix is used.

When you configure a **nameif** command for the interface in a context, the new MAC address is generated immediately. If you enable this command after you configure context interfaces, then MAC addresses are generated for all interfaces immediately after you enter the command. If you use the **no mac-address auto** command, the MAC address for each interface reverts to the default MAC address. For example, subinterfaces of GigabitEthernet 0/1 revert to using the MAC address of GigabitEthernet 0/1.

Note For the MAC address generation method when not using a prefix (not recommended), see the **mac-address auto** command in the *Cisco Security Appliance Command Reference*.

Viewing Assigned MAC Addresses

You can view auto-generated MAC addresses within the system configuration or within the context. This section includes the following topics:

- Viewing MAC Addresses in the System Configuration, page 7-13
- Viewing MAC Addresses Within a Context, page 7-14

Viewing MAC Addresses in the System Configuration

To view the assigned MAC addresses from the system execution space, enter the following command:

```
hostname# show running-config all context [name]
```

The **all** option is required to view the assigned MAC addresses. Although this command is user-configurable in global configuration mode only, the **mac-address auto** command appears as a read-only entry in the configuration for each context along with the assigned MAC address. Only allocated interfaces that are configured with a **nameif** command within the context have a MAC address assigned.

Note If you manually assign a MAC address to an interface, but also have auto-generation enabled, the auto-generated address continues to show in the configuration even though the manual MAC address is the one that is in use. If you later remove the manual MAC address, the auto-generated one shown will be used.

The following output from the **show running-config all context admin** command shows the primary and standby MAC address assigned to the Management0/0 interface:

```
hostname# show running-config all context admin

context admin
  allocate-interface Management0/0
  mac-address auto Management0/0 a24d.0000.1440 a24d.0000.1441
  config-url disk0:/admin.cfg
```

The following output from the **show running-config all context** command shows all the MAC addresses (primary and standby) for all context interfaces. Note that because the GigabitEthernet0/0 and GigabitEthernet0/1 main interfaces are not configured with a **nameif** command inside the contexts, no MAC addresses have been generated for them.

```
hostname# show running-config all context

admin-context admin
context admin
  allocate-interface Management0/0
  mac-address auto Management0/0 a2d2.0400.125a a2d2.0400.125b
  config-url disk0:/admin.cfg
!

context CTX1
  allocate-interface GigabitEthernet0/0
  allocate-interface GigabitEthernet0/0.1-GigabitEthernet0/0.5
  mac-address auto GigabitEthernet0/0.1 a2d2.0400.11bc a2d2.0400.11bd
  mac-address auto GigabitEthernet0/0.2 a2d2.0400.11c0 a2d2.0400.11c1
  mac-address auto GigabitEthernet0/0.3 a2d2.0400.11c4 a2d2.0400.11c5
  mac-address auto GigabitEthernet0/0.4 a2d2.0400.11c8 a2d2.0400.11c9
```

```
  mac-address auto GigabitEthernet0/0.5 a2d2.0400.11cc a2d2.0400.11cd
  allocate-interface GigabitEthernet0/1
  allocate-interface GigabitEthernet0/1.1-GigabitEthernet0/1.3
  mac-address auto GigabitEthernet0/1.1 a2d2.0400.120c a2d2.0400.120d
  mac-address auto GigabitEthernet0/1.2 a2d2.0400.1210 a2d2.0400.1211
  mac-address auto GigabitEthernet0/1.3 a2d2.0400.1214 a2d2.0400.1215
  config-url disk0:/CTX1.cfg
!

context CTX2
  allocate-interface GigabitEthernet0/0
  allocate-interface GigabitEthernet0/0.1-GigabitEthernet0/0.5
  mac-address auto GigabitEthernet0/0.1 a2d2.0400.11ba a2d2.0400.11bb
  mac-address auto GigabitEthernet0/0.2 a2d2.0400.11be a2d2.0400.11bf
  mac-address auto GigabitEthernet0/0.3 a2d2.0400.11c2 a2d2.0400.11c3
  mac-address auto GigabitEthernet0/0.4 a2d2.0400.11c6 a2d2.0400.11c7
  mac-address auto GigabitEthernet0/0.5 a2d2.0400.11ca a2d2.0400.11cb
  allocate-interface GigabitEthernet0/1
  allocate-interface GigabitEthernet0/1.1-GigabitEthernet0/1.3
  mac-address auto GigabitEthernet0/1.1 a2d2.0400.120a a2d2.0400.120b
  mac-address auto GigabitEthernet0/1.2 a2d2.0400.120e a2d2.0400.120f
  mac-address auto GigabitEthernet0/1.3 a2d2.0400.1212 a2d2.0400.1213
  config-url disk0:/CTX2.cfg
!
```

Viewing MAC Addresses Within a Context

To view the MAC address in use by each interface within the context, enter the following command:

```
hostname/context# show interface | include (Interface)|(MAC)
```

For example:

```
hostname/context# show interface | include (Interface)|(MAC)

Interface GigabitEthernet1/1.1 "g1/1.1", is down, line protocol is down
        MAC address a201.0101.0600, MTU 1500
Interface GigabitEthernet1/1.2 "g1/1.2", is down, line protocol is down
        MAC address a201.0102.0600, MTU 1500
Interface GigabitEthernet1/1.3 "g1/1.3", is down, line protocol is down
        MAC address a201.0103.0600, MTU 1500
...
```

Note The **show interface** command shows the MAC address in use; if you manually assign a MAC address and also have auto-generation enabled, then you can only view the unused auto-generated address from within the system configuration.

Changing Between Contexts and the System Execution Space

If you log in to the system execution space (or the admin context using Telnet or SSH), you can change between contexts and perform configuration and monitoring tasks within each context. The running configuration that you edit in a configuration mode, or that is used in the **copy** or **write** commands, depends on your location. When you are in the system execution space, the running configuration consists only of the system configuration; when you are in a context, the running configuration consists only of that context. For example, you cannot view all running configurations (system plus all contexts) by entering the **show running-config** command. Only the current configuration displays.

To change between the system execution space and a context, or between contexts, see the following commands:

- To change to a context, enter the following command:

  ```
  hostname# changeto context name
  ```

  The prompt changes to the following:

  ```
  hostname/name#
  ```

- To change to the system execution space, enter the following command:

  ```
  hostname/admin# changeto system
  ```

  The prompt changes to the following:

  ```
  hostname#
  ```

Managing Security Contexts

This section describes how to manage security contexts, and includes the following topics:

- Removing a Security Context, page 7-15
- Changing the Admin Context, page 7-16
- Changing the Security Context URL, page 7-16
- Reloading a Security Context, page 7-17
- Monitoring Security Contexts, page 7-18

Removing a Security Context

You can only remove a context by editing the system configuration. You cannot remove the current admin context, unless you remove all contexts using the **clear context** command.

Note If you use failover, there is a delay between when you remove the context on the active unit and when the context is removed on the standby unit. You might see an error message indicating that the number of interfaces on the active and standby units are not consistent; this error is temporary and can be ignored.

Use the following commands for removing contexts:

- To remove a single context, enter the following command in the system execution space:

  ```
  hostname(config)# no context name
  ```

  All context commands are also removed.

- To remove all contexts (including the admin context), enter the following command in the system execution space:

  ```
  hostname(config)# clear context
  ```

Changing the Admin Context

The system configuration does not include any network interfaces or network settings for itself; rather, when the system needs to access network resources (such as downloading the contexts from the server), it uses one of the contexts that is designated as the admin context.

The admin context is just like any other context, except that when a user logs in to the admin context, then that user has system administrator rights and can access the system and all other contexts. The admin context is not restricted in any way, and can be used as a regular context. However, because logging into the admin context grants you administrator privileges over all contexts, you might need to restrict access to the admin context to appropriate users.

You can set any context to be the admin context, as long as the configuration file is stored in the internal Flash memory. To set the admin context, enter the following command in the system execution space:

```
hostname(config)# admin-context context_name
```

Any remote management sessions, such as Telnet, SSH, or HTTPS, that are connected to the admin context are terminated. You must reconnect to the new admin context.

Note A few system commands, including **ntp server**, identify an interface name that belongs to the admin context. If you change the admin context, and that interface name does not exist in the new admin context, be sure to update any system commands that refer to the interface.

Changing the Security Context URL

You cannot change the security context URL without reloading the configuration from the new URL.

The security appliance merges the new configuration with the current running configuration. Reentering the same URL also merges the saved configuration with the running configuration. A merge adds any new commands from the new configuration to the running configuration. If the configurations are the same, no changes occur. If commands conflict or if commands affect the running of the context, then the effect of the merge depends on the command. You might get errors, or you might have unexpected results. If the running configuration is blank (for example, if the server was unavailable and the configuration was never downloaded), then the new configuration is used. If you do not want to merge the configurations, you can clear the running configuration, which disrupts any communications through the context, and then reload the configuration from the new URL.

To change the URL for a context, perform the following steps:

Step 1 If you do not want to merge the configuration, change to the context and clear its configuration by entering the following commands. If you want to perform a merge, skip to Step 2.

```
hostname# changeto context name
hostname/name# configure terminal
hostname/name(config)# clear configure all
```

Step 2 If required, change to the system execution space by entering the following command:

```
hostname/name(config)# changeto system
```

Step 3 To enter the context configuration mode for the context you want to change, enter the following command:

```
hostname(config)# context name
```

Step 4 To enter the new URL, enter the following command:

```
hostname(config)# config-url new_url
```

The system immediately loads the context so that it is running.

Reloading a Security Context

You can reload the context in two ways:

- Clear the running configuration and then import the startup configuration.

  This action clears most attributes associated with the context, such as connections and NAT tables.

- Remove the context from the system configuration.

  This action clears additional attributes, such as memory allocation, which might be useful for troubleshooting. However, to add the context back to the system requires you to respecify the URL and interfaces.

This section includes the following topics:

- Reloading by Clearing the Configuration, page 7-17
- Reloading by Removing and Re-adding the Context, page 7-18

Reloading by Clearing the Configuration

To reload the context by clearing the context configuration, and reloading the configuration from the URL, perform the following steps:

Step 1 To change to the context that you want to reload, enter the following command:

```
hostname# changeto context name
```

Step 2 To access configuration mode, enter the following command:

```
hostname/name# configure terminal
```

Step 3 To clear the running configuration, enter the following command:

```
hostname/name(config)# clear configure all
```

This command clears all connections.

Step 4 To reload the configuration, enter the following command:

```
hostname/name(config)# copy startup-config running-config
```

The security appliance copies the configuration from the URL specified in the system configuration. You cannot change the URL from within a context.

Reloading by Removing and Re-adding the Context

To reload the context by removing the context and then re-adding it, perform the steps in the following sections:

1. "Automatically Assigning MAC Addresses to Context Interfaces" section on page 7-11
2. "Configuring a Security Context" section on page 7-7

Monitoring Security Contexts

This section describes how to view and monitor context information, and includes the following topics:

- Viewing Context Information, page 7-18
- Viewing Resource Allocation, page 7-19
- Viewing Resource Usage, page 7-22
- Monitoring SYN Attacks in Contexts, page 7-23

Viewing Context Information

From the system execution space, you can view a list of contexts including the name, allocated interfaces, and configuration file URL.

From the system execution space, view all contexts by entering the following command:

```
hostname# show context [name | detail| count]
```

The **detail** option shows additional information. See the following sample displays below for more information.

If you want to show information for a particular context, specify the *name*.

The **count** option shows the total number of contexts.

The following is sample output from the **show context** command. The following sample display shows three contexts:

```
hostname# show context

Context Name      Interfaces                  URL
*admin            GigabitEthernet0/1.100      disk0:/admin.cfg
                  GigabitEthernet0/1.101
contexta          GigabitEthernet0/1.200      disk0:/contexta.cfg
                  GigabitEthernet0/1.201
contextb          GigabitEthernet0/1.300      disk0:/contextb.cfg
                  GigabitEthernet0/1.301
Total active Security Contexts: 3
```

Table 7-2 shows each field description.

Table 7-2 show context Fields

Field	Description
Context Name	Lists all context names. The context name with the asterisk (*) is the admin context.
Interfaces	The interfaces assigned to the context.
URL	The URL from which the security appliance loads the context configuration.

The following is sample output from the **show context detail** command:

```
hostname# show context detail

Context "admin", has been created, but initial ACL rules not complete
  Config URL: disk0:/admin.cfg
  Real Interfaces: Management0/0
  Mapped Interfaces: Management0/0
  Flags: 0x00000013, ID: 1

Context "ctx", has been created, but initial ACL rules not complete
  Config URL: ctx.cfg
  Real Interfaces: GigabitEthernet0/0.10, GigabitEthernet0/1.20,
     GigabitEthernet0/2.30
  Mapped Interfaces: int1, int2, int3
  Flags: 0x00000011, ID: 2

Context "system", is a system resource
  Config URL: startup-config
  Real Interfaces:
  Mapped Interfaces: Control0/0, GigabitEthernet0/0,
     GigabitEthernet0/0.10, GigabitEthernet0/1, GigabitEthernet0/1.10,
     GigabitEthernet0/1.20, GigabitEthernet0/2, GigabitEthernet0/2.30,
     GigabitEthernet0/3, Management0/0, Management0/0.1
  Flags: 0x00000019, ID: 257

Context "null", is a system resource
  Config URL: ... null ...
  Real Interfaces:
  Mapped Interfaces:
  Flags: 0x00000009, ID: 258
```

See the *Cisco Security Appliance Command Reference* for more information about the **detail** output.

The following is sample output from the **show context count** command:

```
hostname# show context count
Total active contexts: 2
```

Viewing Resource Allocation

From the system execution space, you can view the allocation for each resource across all classes and class members.

To view the resource allocation, enter the following command:

```
hostname# show resource allocation [detail]
```

This command shows the resource allocation, but does not show the actual resources being used. See the "Viewing Resource Usage" section on page 7-22 for more information about actual resource usage.

The **detail** argument shows additional information. See the following sample displays for more information.

The following sample display shows the total allocation of each resource as an absolute value and as a percentage of the available system resources:

```
hostname# show resource allocation
Resource                    Total       % of Avail
 Conns [rate]               35000           N/A
 Inspects [rate]              35000           N/A
 Syslogs [rate]             10500           N/A
 Conns                     305000           30.50%
 Hosts                      78842           N/A
```

```
SSH                                35           35.00%
Telnet                             35           35.00%
Xlates                          91749           N/A
All                         unlimited
```

Table 7-3 shows each field description.

Table 7-3 show resource allocation Fields

Field	Description
Resource	The name of the resource that you can limit.
Total	The total amount of the resource that is allocated across all contexts. The amount is an absolute number of concurrent instances or instances per second. If you specified a percentage in the class definition, the security appliance converts the percentage to an absolute number for this display.
% of Avail	The percentage of the total system resources that is allocated across all contexts, if the resource has a hard system limit. If a resource does not have a system limit, this column shows N/A.

The following is sample output from the **show resource allocation detail** command:

```
hostname# show resource allocation detail
Resource Origin:
    A    Value was derived from the resource 'all'
    C    Value set in the definition of this class
    D    Value set in default class
Resource         Class          Mmbrs  Origin      Limit      Total    Total %
Conns [rate]     default          all      CA  unlimited
                 gold               1       C      34000      34000     N/A
                 silver             1      CA      17000      17000     N/A
                 bronze             0      CA       8500
                 All Contexts:      3                         51000     N/A

Inspects [rate]  default          all      CA  unlimited
                 gold               1      DA  unlimited
                 silver             1      CA      10000      10000     N/A
                 bronze             0      CA       5000
                 All Contexts:      3                         10000     N/A

Syslogs [rate]   default          all      CA  unlimited
                 gold               1       C       6000       6000     N/A
                 silver             1      CA       3000       3000     N/A
                 bronze             0      CA       1500
                 All Contexts:      3                          9000     N/A

Conns            default          all      CA  unlimited
                 gold               1       C     200000     200000     20.00%
                 silver             1      CA     100000     100000     10.00%
                 bronze             0      CA      50000
                 All Contexts:      3                        300000     30.00%

Hosts            default          all      CA  unlimited
                 gold               1      DA  unlimited
                 silver             1      CA      26214      26214      N/A
                 bronze             0      CA      13107
                 All Contexts:      3                         26214      N/A

SSH              default          all       C          5
                 gold               1       D          5          5      5.00%
```

```
                  silver              1      CA          10          10      10.00%
                  bronze              0      CA           5
                  All Contexts:       3                               20      20.00%

Telnet            default           all       C           5
                  gold                1       D           5           5       5.00%
                  silver              1      CA          10          10      10.00%
                  bronze              0      CA           5
                  All Contexts:       3                               20      20.00%

Xlates            default           all      CA   unlimited
                  gold                1      DA   unlimited
                  silver              1      CA       23040       23040      N/A
                  bronze              0      CA       11520
                  All Contexts:       3                           23040      N/A

mac-addresses     default           all       C       65535
                  gold                1       D       65535       65535    100.00%
                  silver              1      CA        6553        6553      9.99%
                  bronze              0      CA        3276
                  All Contexts:       3                          137623    209.99%
```

Table 7-4 shows each field description.

Table 7-4 show resource allocation detail Fields

Field	Description
Resource	The name of the resource that you can limit.
Class	The name of each class, including the default class. The All contexts field shows the total values across all classes.
Mmbrs	The number of contexts assigned to each class.
Origin	The origin of the resource limit, as follows: • A—You set this limit with the **all** option, instead of as an individual resource. • C—This limit is derived from the member class. • D—This limit was not defined in the member class, but was derived from the default class. For a context assigned to the default class, the value will be “C” instead of “D.” The security appliance can combine “A” with “C” or “D.”
Limit	The limit of the resource per context, as an absolute number. If you specified a percentage in the class definition, the security appliance converts the percentage to an absolute number for this display.
Total	The total amount of the resource that is allocated across all contexts in the class. The amount is an absolute number of concurrent instances or instances per second. If the resource is unlimited, this display is blank.
% of Avail	The percentage of the total system resources that is allocated across all contexts in the class. If the resource is unlimited, this display is blank. If the resource does not have a system limit, then this column shows N/A.

Viewing Resource Usage

From the system execution space, you can view the resource usage for each context and display the system resource usage.

From the system execution space, view the resource usage for each context by entering the following command:

```
hostname# show resource usage [context context_name | top n | all | summary | system]
[resource {resource_name | all} | detail] [counter counter_name [count_threshold]]
```

By default, **all** context usage is displayed; each context is listed separately.

Enter the **top** *n* keyword to show the contexts that are the top *n* users of the specified resource. You must specify a single resource type, and not **resource all**, with this option.

The **summary** option shows all context usage combined.

The **system** option shows all context usage combined, but shows the system limits for resources instead of the combined context limits.

For the **resource** *resource_name*, see Table 7-1 for available resource names. See also the **show resource type** command. Specify **all** (the default) for all types.

The **detail** option shows the resource usage of all resources, including those you cannot manage. For example, you can view the number of TCP intercepts.

The **counter** *counter_name* is one of the following keywords:

- **current**—Shows the active concurrent instances or the current rate of the resource.
- **denied**—Shows the number of instances that were denied because they exceeded the resource limit shown in the Limit column.
- **peak**—Shows the peak concurrent instances, or the peak rate of the resource since the statistics were last cleared, either using the **clear resource usage** command or because the device rebooted.
- **all**—(Default) Shows all statistics.

The *count_threshold* sets the number above which resources are shown. The default is 1. If the usage of the resource is below the number you set, then the resource is not shown. If you specify **all** for the counter name, then the *count_threshold* applies to the current usage.

Note To show all resources, set the *count_threshold* to **0**.

The following is sample output from the **show resource usage context** command, which shows the resource usage for the admin context:

```
hostname# show resource usage context admin

Resource              Current         Peak      Limit    Denied  Context
Telnet                      1            1          5         0  admin
Conns                      44           55        N/A         0  admin
Hosts                      45           56        N/A         0  admin
```

The following is sample output from the **show resource usage summary** command, which shows the resource usage for all contexts and all resources. This sample shows the limits for 6 contexts.

```
hostname# show resource usage summary

Resource              Current         Peak      Limit     Denied Context
Syslogs [rate]           1743         2132      N/A            0 Summary
Conns                     584          763      280000(S)      0 Summary
```

```
Xlates                    8526         8966      N/A            0 Summary
Hosts                      254          254      N/A            0 Summary
Conns [rate]               270          535      N/A         1704 Summary
Inspects [rate]            270          535      N/A            0 Summary
S = System: Combined context limits exceed the system limit; the system limit is shown.
```

The following is sample output from the **show resource usage summary** command, which shows the limits for 25 contexts. Because the context limit for Telnet and SSH connections is 5 per context, then the combined limit is 125. The system limit is only 100, so the system limit is shown.

```
hostname# show resource usage summary

Resource              Current         Peak      Limit    Denied   Context
Telnet                      1            1        100[S]      0   Summary
SSH                         2            2        100[S]      0   Summary
Conns                      56           90        N/A         0   Summary
Hosts                      89          102        N/A         0   Summary
S = System: Combined context limits exceed the system limit; the system limit is shown.
```

The following is sample output from the **show resource usage system** command, which shows the resource usage for all contexts, but it shows the system limit instead of the combined context limits. The **counter all 0** option is used to show resources that are not currently in use. The Denied statistics indicate how many times the resource was denied due to the system limit, if available.

```
hostname# show resource usage system counter all 0

Resource              Current         Peak      Limit        Denied    Context
Telnet                      0            0        100             0    System
SSH                         0            0        100             0    System
ASDM                        0            0         32             0    System
Syslogs [rate]              1           18        N/A             0    System
Conns                       0            1     280000             0    System
Xlates                      0            0        N/A             0    System
Hosts                       0            2        N/A             0    System
Conns [rate]                1            1        N/A             0    System
Inspects [rate]             0            0        N/A             0    System
```

Monitoring SYN Attacks in Contexts

The security appliance prevents SYN attacks using TCP Intercept. TCP Intercept uses the SYN cookies algorithm to prevent TCP SYN-flooding attacks. A SYN-flooding attack consists of a series of SYN packets usually originating from spoofed IP addresses. The constant flood of SYN packets keeps the server SYN queue full, which prevents it from servicing connection requests. When the embryonic connection threshold of a connection is crossed, the security appliance acts as a proxy for the server and generates a SYN-ACK response to the client SYN request. When the security appliance receives an ACK back from the client, it can then authenticate the client and allow the connection to the server.

You can monitor the rate of attacks for individual contexts using the **show perfmon** command; you can monitor the amount of resources being used by TCP intercept for individual contexts using the **show resource usage detail** command; you can monitor the resources being used by TCP intercept for the entire system using the **show resource usage summary detail** command.

The following is sample output from the **show perfmon** command that shows the rate of TCP intercepts for a context called admin.

```
hostname/admin# show perfmon

Context:admin
PERFMON STATS:   Current      Average
Xlates               0/s          0/s
```

```
Connections           0/s           0/s
TCP Conns             0/s           0/s
UDP Conns             0/s           0/s
URL Access            0/s           0/s
URL Server Req        0/s           0/s
WebSns Req            0/s           0/s
TCP Fixup             0/s           0/s
HTTP Fixup            0/s           0/s
FTP Fixup             0/s           0/s
AAA Authen            0/s           0/s
AAA Author            0/s           0/s
AAA Account           0/s           0/s
TCP Intercept    322779/s      322779/s
```

The following is sample output from the **show resource usage detail** command that shows the amount of resources being used by TCP Intercept for individual contexts. (Sample text in italics shows the TCP intercept information.)

```
hostname(config)# show resource usage detail
Resource              Current         Peak      Limit        Denied Context
memory                 843732       847288  unlimited             0 admin
chunk:channels             14           15  unlimited             0 admin
chunk:fixup                15           15  unlimited             0 admin
chunk:hole                  1            1  unlimited             0 admin
chunk:ip-users             10           10  unlimited             0 admin
chunk:list-elem            21           21  unlimited             0 admin
chunk:list-hdr              3            4  unlimited             0 admin
chunk:route                 2            2  unlimited             0 admin
chunk:static                1            1  unlimited             0 admin
tcp-intercepts         328787       803610  unlimited             0 admin
np-statics                  3            3  unlimited             0 admin
statics                     1            1  unlimited             0 admin
ace-rules                   1            1  unlimited             0 admin
console-access-rul          2            2  unlimited             0 admin
fixup-rules                14           15  unlimited             0 admin
memory                 959872       960000  unlimited             0 c1
chunk:channels             15           16  unlimited             0 c1
chunk:dbgtrace              1            1  unlimited             0 c1
chunk:fixup                15           15  unlimited             0 c1
chunk:global                1            1  unlimited             0 c1
chunk:hole                  2            2  unlimited             0 c1
chunk:ip-users             10           10  unlimited             0 c1
chunk:udp-ctrl-blk          1            1  unlimited             0 c1
chunk:list-elem            24           24  unlimited             0 c1
chunk:list-hdr              5            6  unlimited             0 c1
chunk:nat                   1            1  unlimited             0 c1
chunk:route                 2            2  unlimited             0 c1
chunk:static                1            1  unlimited             0 c1
tcp-intercept-rate      16056        16254  unlimited             0 c1
globals                     1            1  unlimited             0 c1
np-statics                  3            3  unlimited             0 c1
statics                     1            1  unlimited             0 c1
nats                        1            1  unlimited             0 c1
ace-rules                   2            2  unlimited             0 c1
console-access-rul          2            2  unlimited             0 c1
fixup-rules                14           15  unlimited             0 c1
memory              232695716    232020648  unlimited             0 system
chunk:channels             17           20  unlimited             0 system
chunk:dbgtrace              3            3  unlimited             0 system
chunk:fixup                15           15  unlimited             0 system
chunk:ip-users              4            4  unlimited             0 system
chunk:list-elem          1014         1014  unlimited             0 system
chunk:list-hdr              1            1  unlimited             0 system
chunk:route                 1            1  unlimited             0 system
```

```
block:16384                510          885  unlimited              0 system
block:2048                  32           34  unlimited              0 system
```

The following sample output shows the resources being used by TCP intercept for the entire system. (Sample text in italics shows the TCP intercept information.)

```
hostname(config)# show resource usage summary detail
Resource              Current         Peak      Limit         Denied Context
memory              238421312    238434336  unlimited              0 Summary
chunk:channels              46           48  unlimited              0 Summary
chunk:dbgtrace               4            4  unlimited              0 Summary
chunk:fixup                 45           45  unlimited              0 Summary
chunk:global                 1            1  unlimited              0 Summary
chunk:hole                   3            3  unlimited              0 Summary
chunk:ip-users              24           24  unlimited              0 Summary
chunk:udp-ctrl-blk           1            1  unlimited              0 Summary
chunk:list-elem           1059         1059  unlimited              0 Summary
chunk:list-hdr              10           11  unlimited              0 Summary
chunk:nat                    1            1  unlimited              0 Summary
chunk:route                  5            5  unlimited              0 Summary
chunk:static                 2            2  unlimited              0 Summary
block:16384                510          885  unlimited              0 Summary
block:2048                  32           35  unlimited              0 Summary
tcp-intercept-rate      341306       811579  unlimited              0 Summary
globals                      1            1  unlimited              0 Summary
np-statics                   6            6  unlimited              0 Summary
statics                      2            2        N/A              0 Summary
nats                         1            1        N/A              0 Summary
ace-rules                    3            3        N/A              0 Summary
console-access-rul           4            4        N/A              0 Summary
fixup-rules                 43           44        N/A              0 Summary
```



CHAPTER 8

Configuring Interface Parameters

This chapter describes how to configure each interface (physical, redundant, or subinterface) for a name, security level, and IP address.

- For single context mode, the procedures in this chapter continue the interface configuration started in Chapter 6, "Configuring Ethernet Settings, Redundant Interfaces, and Subinterfaces."
- For multiple context mode, the procedures in Chapter 6, "Configuring Ethernet Settings, Redundant Interfaces, and Subinterfaces," are performed in the system execution space, while the procedures in this chapter are performed within each security context.

Note To configure interfaces for the ASA 5505 adaptive security appliance, see Chapter 5, "Configuring Switch Ports and VLAN Interfaces for the Cisco ASA 5505 Adaptive Security Appliance."

This chapter includes the following sections:

- Security Level Overview, page 8-1
- Configuring Interface Parameters, page 8-2
- Allowing Communication Between Interfaces on the Same Security Level, page 8-7

Security Level Overview

Each interface must have a security level from 0 (lowest) to 100 (highest). For example, you should assign your most secure network, such as the inside host network, to level 100. While the outside network connected to the Internet can be level 0. Other networks, such as DMZs can be in between. You can assign interfaces to the same security level. See the "Allowing Communication Between Interfaces on the Same Security Level" section on page 8-7 for more information.

The level controls the following behavior:

- Network access—By default, there is an implicit permit from a higher security interface to a lower security interface (outbound). Hosts on the higher security interface can access any host on a lower security interface. You can limit access by applying an access list to the interface.

  If you enable communication for same security interfaces (see the "Allowing Communication Between Interfaces on the Same Security Level" section on page 8-7), there is an implicit permit for interfaces to access other interfaces on the same security level or lower.
- Inspection engines—Some application inspection engines are dependent on the security level. For same security interfaces, inspection engines apply to traffic in either direction.

  - NetBIOS inspection engine—Applied only for outbound connections.
  - SQL*Net inspection engine—If a control connection for the SQL*Net (formerly OraServ) port exists between a pair of hosts, then only an inbound data connection is permitted through the security appliance.
- Filtering—HTTP(S) and FTP filtering applies only for outbound connections (from a higher level to a lower level).

  If you enable communication for same security interfaces, you can filter traffic in either direction.
- NAT control—When you enable NAT control, you must configure NAT for hosts on a higher security interface (inside) when they access hosts on a lower security interface (outside).

  Without NAT control, or for same security interfaces, you can choose to use NAT between any interface, or you can choose not to use NAT. Keep in mind that configuring NAT for an outside interface might require a special keyword.
- **established** command—This command allows return connections from a lower security host to a higher security host if there is already an established connection from the higher level host to the lower level host.

  If you enable communication for same security interfaces , you can configure **established** commands for both directions.

Configuring Interface Parameters

Before you can complete your configuration and allow traffic through the security appliance, you need to configure an interface name, and for routed mode, an IP address.

Note If you are using failover, do not use this procedure to name interfaces that you are reserving for failover and Stateful Failover communications. See Chapter 15, "Configuring Failover." to configure the failover and state links.

This section includes the following topics:

- Interface Parameters Overview, page 8-2
- Configuring the Interface, page 8-3

Interface Parameters Overview

This section describes interface parameters and includes the following topics:

- Default State of Interfaces, page 8-3
- Default Security Level, page 8-3
- Multiple Context Mode Guidelines, page 8-3

Default State of Interfaces

The default state of an interface depends on the type and the context mode.

In multiple context mode, all allocated interfaces are enabled by default, no matter what the state of the interface is in the system execution space. However, for traffic to pass through the interface, the interface also has to be enabled in the system execution space. If you shut down an interface in the system execution space, then that interface is down in all contexts that share it.

In single mode or in the system execution space, interfaces have the following default states:

- Physical interfaces—Disabled.
- Redundant Interfaces—Enabled. However, for traffic to pass through the redundant interface, the member physical interfaces must also be enabled.
- Subinterfaces—Enabled. However, for traffic to pass through the subinterface, the physical interface must also be enabled.

Default Security Level

The default security level is 0. If you name an interface "inside" and you do not set the security level explicitly, then the security appliance sets the security level to 100.

Note If you change the security level of an interface, and you do not want to wait for existing connections to time out before the new security information is used, you can clear the connections using the **clear local-host** command.

Multiple Context Mode Guidelines

For multiple context mode, follow these guidelines:

- Configure the context interfaces from within each context.
- Configure context interfaces that you already assigned to the context in the system configuration. Other interfaces are not available.
- Configure Ethernet settings, redundant interfaces, and subinterfaces in the system configuration. No other configuration is available. The exception is for failover interfaces, which are configured in the system configuration. Do not configure failover interfaces with the procedures in this chapter. See Chapter 15, "Configuring Failover," for more information.

Configuring the Interface

To configure an interface or subinterface, perform the following steps:

Step 1 To specify the interface you want to configure, enter the following command:

```
hostname(config)# interface {{redundant number| physical_interface}[.subinterface] |
mapped_name}
hostname(config-if)#
```

The **redundant** *number* argument is the redundant interface ID, such as **redundant 1**.

Append the *subinterface* ID to the physical or redundant interface ID separated by a period (.).

In multiple context mode, enter the *mapped_name* if one was assigned using the **allocate-interface** command.

The *physical_interface* ID includes the type, slot, and port number as *type* [*slot*/]*port*. The physical interface types include the following:

- **ethernet**
- **gigabitethernet**
- **management** (ASA 5500 only)

For the PIX 500 series security appliance, enter the type followed by the port number, for example, **ethernet 0**.

For the ASA 5500 series adaptive security appliance, enter the type followed by *slot*/*port*, for example, **gigabitethernet 0/1** or **ethernet 0/1**.

Note For the ASA 5550 adaptive security appliance, for maximum throughput, be sure to balance your traffic over the two interface slots; for example, assign the inside interface to slot 1 and the outside interface to slot 0.

The ASA 5500 management interface is a Fast Ethernet interface designed for management traffic only, and is specified as **management 0/0**. You can, however, use it for through traffic if desired (see the **management-only** command). In transparent firewall mode, you can use the management interface (for management purposes) in addition to the two interfaces allowed for through traffic. You can also add subinterfaces to the management interface to provide management in each security context for multiple context mode.

For example, enter the following command:

```
hostname(config)# interface gigabitethernet 0/1.1
```

Step 2 To name the interface, enter the following command:

```
hostname(config-if)# nameif name
```

The *name* is a text string up to 48 characters, and is not case-sensitive. You can change the name by reentering this command with a new value. Do not enter the **no** form, because that command causes all commands that refer to that name to be deleted.

Step 3 To set the security level, enter the following command:

```
hostname(config-if)# security-level number
```

Where *number* is an integer between 0 (lowest) and 100 (highest).

Step 4 (Optional) To set an interface to management-only mode, enter the following command:

```
hostname(config-if)# management-only
```

The ASA 5510 and higher adaptive security appliance includes a dedicated management interface called Management 0/0, which is meant to support traffic to the security appliance. However, you can configure any interface to be a management-only interface using the **management-only** command. Also, for Management 0/0, you can disable management-only mode so the interface can pass through traffic just like any other interface.

Note Transparent firewall mode allows only two interfaces to pass through traffic; however, on the ASA 5510 and higher adaptive security appliance, you can use the Management 0/0 interface (either the physical interface or a subinterface) as a third interface for management traffic. The mode is not configurable in this case and must always be management-only.

Step 5 To set the IP address, enter one of the following commands.

In routed firewall mode, set the IP address for all interfaces. In transparent firewall mode, do not set the IP address for each interface, but rather set it for the whole security appliance or context. The exception is for the Management 0/0 management-only interface, which does not pass through traffic. To set the transparent firewall mode whole security appliance or context management IP address, see the "Setting the Management IP Address for a Transparent Firewall" section on page 9-5. To set the IP address of the Management 0/0 interface or subinterface, use one of the following commands.

To set an IPv6 address, see the "Configuring IPv6 on an Interface" section on page 13-3.

For use with failover, you must set the IP address and standby address manually; DHCP and PPPoE are not supported.

- To set the IP address manually, enter the following command:

  ```
  hostname(config-if)# ip address ip_address [mask] [standby ip_address]
  ```

  where the *ip_address* and *mask* arguments set the interface IP address and subnet mask.

  The **standby** *ip_address* argument is used for failover. See Chapter 15, "Configuring Failover," for more information.

- To obtain an IP address from a DHCP server, enter the following command:

  ```
  hostname(config-if)# ip address dhcp [setroute]
  ```

  where the **setroute** keyword lets the security appliance use the default route supplied by the DHCP server.

  Reenter this command to reset the DHCP lease and request a new lease.

  If you do not enable the interface using the **no shutdown** command before you enter the **ip address dhcp** command, some DHCP requests might not be sent.

- To obtain an IP address from a PPPoE server, see Chapter 37, "Configuring the PPPoE Client."

  PPPoE is not supported in multiple context mode.

Step 6 (Optional) To assign a private MAC address to this interface, enter the following command:

```
hostname(config-if)# mac-address mac_address [standby mac_address]
```

The *mac_address* is in H.H.H format, where H is a 16-bit hexadecimal digit. For example, the MAC address 00-0C-F1-42-4C-DE is entered as 000C.F142.4CDE.

By default, the physical interface uses the burned-in MAC address, and all subinterfaces of a physical interface use the same burned-in MAC address. A redundant interface uses the MAC address of the first physical interface that you add. If you change the order of the member interfaces in the configuration, then the MAC address changes to match the MAC address of the interface that is now listed first. If you assign a MAC address to the redundant interface using this command, then it is used regardless of the member interface MAC addresses.

In multiple context mode, if you share an interface between contexts, you can assign a unique MAC address to the interface in each context. This feature lets the security appliance easily classify packets into the appropriate context. Using a shared interface without unique MAC addresses is possible, but has some limitations. See the "How the Security Appliance Classifies Packets" section on page 4-3 for more

information. You can assign each MAC address manually, or you can automatically generate MAC addresses for shared interfaces in contexts. See the "Automatically Assigning MAC Addresses to Context Interfaces" section on page 7-11 to automatically generate MAC addresses. If you automatically generate MAC addresses, you can use the **mac-address** command to override the generated address. The first two bytes of a manual MAC address cannot be A2 if you also want to use auto-generated MAC addresses.

For single context mode, or for interfaces that are not shared in multiple context mode, you might want to assign unique MAC addresses to subinterfaces. For example, your service provider might perform access control based on the MAC address.

For use with failover, set the **standby** MAC address. If the active unit fails over and the standby unit becomes active, the new active unit starts using the active MAC addresses to minimize network disruption, while the old active unit uses the standby address.

Step 7 To enable the interface, if it is not already enabled, enter the following command:

```
hostname(config-if)# no shutdown
```

To disable the interface, enter the **shutdown** command. If you enter the **shutdown** command for a physical or redundant interface, you also shut down all subinterfaces. If you shut down an interface in the system execution space, then that interface is shut down in all contexts that share it, even though the context configurations show the interface as enabled.

The following example configures parameters for the physical interface in single mode:

```
hostname(config)# interface gigabitethernet 0/1
hostname(config-if)# speed 1000
hostname(config-if)# duplex full
hostname(config-if)# nameif inside
hostname(config-if)# security-level 100
hostname(config-if)# ip address 10.1.1.1 255.255.255.0
hostname(config-if)# no shutdown
```

The following example configures parameters for a subinterface in single mode:

```
hostname(config)# interface gigabitethernet 0/1.1
hostname(config-subif)# vlan 101
hostname(config-subif)# nameif dmz1
hostname(config-subif)# security-level 50
hostname(config-subif)# ip address 10.1.2.1 255.255.255.0
hostname(config-subif)# mac-address 000C.F142.4CDE standby 020C.F142.4CDE
hostname(config-subif)# no shutdown
```

The following example configures interface parameters in multiple context mode for the system configuration, and allocates the gigabitethernet 0/1.1 subinterface to contextA:

```
hostname(config)# interface gigabitethernet 0/1
hostname(config-if)# speed 1000
hostname(config-if)# duplex full
hostname(config-if)# no shutdown
hostname(config-if)# interface gigabitethernet 0/1.1
hostname(config-subif)# vlan 101
hostname(config-subif)# no shutdown
hostname(config-subif)# context contextA
hostname(config-ctx)# ...
hostname(config-ctx)# allocate-interface gigabitethernet 0/1.1
```

The following example configures parameters in multiple context mode for the context configuration:

```
hostname/contextA(config)# interface gigabitethernet 0/1.1
```

```
hostname/contextA(config-if)# nameif inside
hostname/contextA(config-if)# security-level 100
hostname/contextA(config-if)# ip address 10.1.2.1 255.255.255.0
hostname/contextA(config-if)# mac-address 030C.F142.4CDE standby 040C.F142.4CDE
hostname/contextA(config-if)# no shutdown
```

Allowing Communication Between Interfaces on the Same Security Level

By default, interfaces on the same security level cannot communicate with each other. Allowing communication between same security interfaces provides the following benefits:

- You can configure more than 101 communicating interfaces.

  If you use different levels for each interface and do not assign any interfaces to the same security level, you can configure only one interface per level (0 to 100).
- You want traffic to flow freely between all same security interfaces without access lists.

Note If you enable NAT control, you do not need to configure NAT between same security level interfaces. See the "NAT and Same Security Level Interfaces" section on page 19-15 for more information on NAT and same security level interfaces.

If you enable same security interface communication, you can still configure interfaces at different security levels as usual.

To enable interfaces on the same security level so that they can communicate with each other, enter the following command:

```
hostname(config)# same-security-traffic permit inter-interface
```

To disable this setting, use the **no** form of this command.



CHAPTER 9

Configuring Basic Settings

This chapter describes how to configure basic settings on your security appliance that are typically required for a functioning configuration. This chapter includes the following sections:

- Changing the Login Password, page 9-1
- Changing the Enable Password, page 9-1
- Setting the Hostname, page 9-2
- Setting the Domain Name, page 9-2
- Setting the Date and Time, page 9-2
- Setting the Management IP Address for a Transparent Firewall, page 9-5

Changing the Login Password

The login password is used for Telnet and SSH connections. By default, the login password is "cisco." To change the password, enter the following command:

```
hostname(config)# {passwd | password} password
```

You can enter **passwd** or **password**. The password is a case-sensitive password of up to 16 alphanumeric and special characters. You can use any character in the password except a question mark or a space.

The password is saved in the configuration in encrypted form, so you cannot view the original password after you enter it. Use the no password command to restore the password to the default setting.

Changing the Enable Password

The enable password lets you enter privileged EXEC mode. By default, the enable password is blank. To change the enable password, enter the following command:

```
hostname(config)# enable password password
```

The *password* is a case-sensitive password of up to 16 alphanumeric and special characters. You can use any character in the password except a question mark or a space.

This command changes the password for the highest privilege level. If you configure local command authorization, you can set enable passwords for each privilege level from 0 to 15.

The password is saved in the configuration in encrypted form, so you cannot view the original password after you enter it. Enter the **enable password** command without a password to set the password to the default, which is blank.

Setting the Hostname

When you set a hostname for the security appliance, that name appears in the command line prompt. If you establish sessions to multiple devices, the hostname helps you keep track of where you enter commands. The default hostname depends on your platform.

For multiple context mode, the hostname that you set in the system execution space appears in the command line prompt for all contexts. The hostname that you optionally set within a context does not appear in the command line, but can be used by the **banner** command **$(hostname)** token.

To specify the hostname for the security appliance or for a context, enter the following command:

```
hostname(config)# hostname name
```

This name can be up to 63 characters. A hostname must start and end with a letter or digit, and have as interior characters only letters, digits, or a hyphen.

This name appears in the command line prompt. For example:

```
hostname(config)# hostname farscape
farscape(config)#
```

Setting the Domain Name

The security appliance appends the domain name as a suffix to unqualified names. For example, if you set the domain name to "example.com," and specify a syslog server by the unqualified name of "jupiter," then the security appliance qualifies the name to "jupiter.example.com."

The default domain name is default.domain.invalid.

For multiple context mode, you can set the domain name for each context, as well as within the system execution space.

To specify the domain name for the security appliance, enter the following command:

```
hostname(config)# domain-name name
```

For example, to set the domain as example.com, enter the following command:

```
hostname(config)# domain-name example.com
```

Setting the Date and Time

This section describes how to set the date and time, either manually or dynamically using an NTP server. Time derived from an NTP server overrides any time set manually. This section also describes how to set the time zone and daylight saving time date range.

Note In multiple context mode, set the time in the system configuration only.

This section includes the following topics:

- Setting the Time Zone and Daylight Saving Time Date Range, page 9-3
- Setting the Date and Time Using an NTP Server, page 9-4
- Setting the Date and Time Manually, page 9-4

Setting the Time Zone and Daylight Saving Time Date Range

By default, the time zone is UTC and the daylight saving time date range is from 2:00 a.m. on the first Sunday in April to 2:00 a.m. on the last Sunday in October. To change the time zone and daylight saving time date range, perform the following steps:

Step 1 To set the time zone, enter the following command in global configuration mode:

```
hostname(config)# clock timezone zone [-]hours [minutes]
```

Where *zone* specifies the time zone as a string, for example, **PST** for Pacific Standard Time.

The [**-**]*hours* value sets the number of hours of offset from UTC. For example, PST is **-8** hours.

The *minutes* value sets the number of minutes of offset from UTC.

Step 2 To change the date range for daylight saving time from the default, enter one of the following commands.

The default recurring date range is from 2:00 a.m. on the second Sunday in March to 2:00 a.m. on the first Sunday in November.

- To set the start and end dates for daylight saving time as a specific date in a specific year, enter the following command:

  ```
  hostname(config)# clock summer-time zone date {day month | month day} year hh:mm {day month | month day} year hh:mm [offset]
  ```

  If you use this command, you need to reset the dates every year.

  The *zone* value specifies the time zone as a string, for example, **PDT** for Pacific Daylight Time.

  The *day* value sets the day of the month, from 1 to 31. You can enter the day and month as **April 1** or as **1 April**, for example, depending on your standard date format.

  The *month* value sets the month as a string. You can enter the day and month as **April 1** or as **1 April**, for example, depending on your standard date format.

  The *year* value sets the year using four digits, for example, **2004**. The year range is 1993 to 2035.

  The *hh:mm* value sets the hour and minutes in 24-hour time.

  The *offset* value sets the number of minutes to change the time for daylight saving time. By default, the value is 60 minutes.

- To specify the start and end dates for daylight saving time, in the form of a day and time of the month, and not a specific date in a year, enter the following command.

  ```
  hostname(config)# clock summer-time zone recurring [week weekday month hh:mm week weekday month hh:mm] [offset]
  ```

  This command lets you set a recurring date range that you do not need to alter yearly.

  The *zone* value specifies the time zone as a string, for example, **PDT** for Pacific Daylight Time.

  The *week* value specifies the week of the month as an integer between 1 and 4 or as the words **first** or **last**. For example, if the day might fall in the partial fifth week, then specify **last**.

The *weekday* value specifies the day of the week: **Monday**, **Tuesday**, **Wednesday**, and so on.

The *month* value sets the month as a string.

The *hh:mm* value sets the hour and minutes in 24-hour time.

The *offset* value sets the number of minutes to change the time for daylight saving time. By default, the value is 60 minutes.

Setting the Date and Time Using an NTP Server

To obtain the date and time from an NTP server, perform the following steps:

Step 1 To configure authentication with an NTP server, perform the following steps:

a. To enable authentication, enter the following command:

```
hostname(config)# ntp authenticate
```

b. To specify an authentication key ID to be a trusted key, which is required for authentication with an NTP server, enter the following command:

```
hostname(config)# ntp trusted-key key_id
```

Where the *key_id* is between 1 and 4294967295. You can enter multiple trusted keys for use with multiple servers.

c. To set a key to authenticate with an NTP server, enter the following command:

```
hostname(config)# ntp authentication-key key_id md5 key
```

Where *key_id* is the ID you set in Step 1b using the **ntp trusted-key** command, and key is a string up to 32 characters in length.

Step 2 To identify an NTP server, enter the following command:

```
hostname(config)# ntp server ip_address [key key_id] [source interface_name] [prefer]
```

Where the *key_id* is the ID you set in Step 1b using the **ntp trusted-key** command.

The **source** *interface_name* identifies the outgoing interface for NTP packets if you do not want to use the default interface in the routing table. Because the system does not include any interfaces in multiple context mode, specify an interface name defined in the admin context.

The **prefer** keyword sets this NTP server as the preferred server if multiple servers have similar accuracy. NTP uses an algorithm to determine which server is the most accurate and synchronizes to that one. If servers are of similar accuracy, then the **prefer** keyword specifies which of those servers to use. However, if a server is significantly more accurate than the preferred one, the security appliance uses the more accurate one. For example, the security appliance uses a server of stratum 2 over a server of stratum 3 that is preferred.

You can identify multiple servers; the security appliance uses the most accurate server.

Setting the Date and Time Manually

To set the date time manually, enter the following command:

```
hostname# clock set hh:mm:ss {month day | day month} year
```

Where *hh*:*mm*:*ss* sets the hour, minutes, and seconds in 24-hour time. For example, set **20:54:00** for 8:54 pm.

The *day* value sets the day of the month, from 1 to 31. You can enter the day and month as **april 1** or as **1 april**, for example, depending on your standard date format.

The *month* value sets the month. Depending on your standard date format, you can enter the day and month as **april 1** or as **1 april**.

The *year* value sets the year using four digits, for example, **2004**. The year range is 1993 to 2035.

The default time zone is UTC. If you change the time zone after you enter the **clock set** command using the **clock timezone** command, the time automatically adjusts to the new time zone.

This command sets the time in the hardware chip, and does not save the time in the configuration file. This time endures reboots. Unlike the other **clock** commands, this command is a privileged EXEC command. To reset the clock, you need to set a new time for the **clock set** command.

Setting the Management IP Address for a Transparent Firewall

Transparent firewall mode only

A transparent firewall does not participate in IP routing. The only IP configuration required for the security appliance is to set the management IP address. This address is required because the security appliance uses this address as the source address for traffic originating on the security appliance, such as system messages or communications with AAA servers. You can also use this address for remote management access.

For multiple context mode, set the management IP address within each context.

To set the management IP address, enter the following command:

```
hostname(config)# ip address ip_address [mask] [standby ip_address]
```

This address must be on the same subnet as the upstream and downstream routers. You cannot set the subnet to a host subnet (255.255.255.255). This address must be IPv4; the transparent firewall does not support IPv6.

The **standby** keyword and address is used for failover. See Chapter 15, “Configuring Failover,” for more information.



CHAPTER 10

Configuring IP Routing

This chapter describes how to configure IP routing on the security appliance. This chapter includes the following sections:

- How Routing Behaves Within the ASA Security Appliance, page 10-1
- Configuring Static and Default Routes, page 10-2
- Defining Route Maps, page 10-7
- Configuring OSPF, page 10-8
- Configuring RIP, page 10-20
- Configuring EIGRP, page 10-24
- The Routing Table, page 10-33
- Dynamic Routing and Failover, page 10-36

How Routing Behaves Within the ASA Security Appliance

The ASA security appliance uses both routing table and XLATE tables for routing decisions. To handle destination IP translated traffic, that is, untranslated traffic, ASA searches for existing XLATE, or static translation to select the egress interface. The selection process is as follows:

Egress Interface Selection Process

1. If destination IP translating XLATE already exists, the egress interface for the packet is determined from the XLATE table, but not from the routing table.
2. If destination IP translating XLATE does not exist, but a matching static translation exists, then the egress interface is determined from the static route and an XLATE is created, and the routing table is not used.
3. If destination IP translating XLATE does not exist and no matching static translation exists, the packet is not destination IP translated. The security appliance processes this packet by looking up the route to select egress interface, then source IP translation is performed (if necessary).

   For regular dynamic outbound NAT, initial outgoing packets are routed using the route table and then creating the XLATE. Incoming return packets are forwarded using existing XLATE only. For static NAT, destination translated incoming packets are always forwarded using existing XLATE or static translation rules.

Next Hop Selection Process

After selecting egress interface using any method described above, an additional route lookup is performed to find out suitable next hop(s) that belong to previously selected egress interface. If there are no routes in routing table that explicitly belong to selected interface, the packet is dropped with level 6 error message 110001 "`no route to host`", even if there is another route for a given destination network that belongs to different egress interface. If the route that belongs to selected egress interface is found, the packet is forwarded to corresponding next hop.

Load sharing on the security appliance is possible only for multiple next-hops available using single egress interface. Load sharing cannot share multiple egress interfaces.

If dynamic routing is in use on security appliance and route table changes after XLATE creation, for example route flap, then destination translated traffic is still forwarded using old XLATE, not via route table, until XLATE times out. It may be either forwarded to wrong interface or dropped with message 110001 "`no route to host`" if old route was removed from the old interface and attached to another one by routing process.

The same problem may happen when there is no route flaps on the security appliance itself, but some routing process is flapping around it, sending source translated packets that belong to the same flow through the security appliance using different interfaces. Destination translated return packets may be forwarded back using the wrong egress interface.

This issue has a high probability in same security traffic configuration, where virtually any traffic may be either source-translated or destination-translated, depending on direction of initial packet in the flow. When this issue occurs after a route flap, it can be resolved manually by using the **`clear xlate`** command, or automatically resolved by an XLATE timeout. XLATE timeout may be decreased if necessary. To ensure that this rarely happens, make sure that there is no route flaps on security appliance and around it. That is, ensure that destination translated packets that belong to the same flow are always forwarded the same way through the security appliance.

Configuring Static and Default Routes

This section describes how to configure static and default routes on the security appliance.

Multiple context mode does not support dynamic routing, so you must use static routes for any networks to which the security appliance is not directly connected; for example, when there is a router between a network and the security appliance.

You might want to use static routes in single context mode in the following cases:

- Your networks use a different router discovery protocol from RIP or OSPF.
- Your network is small and you can easily manage static routes.
- You do not want the traffic or CPU overhead associated with routing protocols.

The simplest option is to configure a default route to send all traffic to an upstream router, relying on the router to route the traffic for you. However, in some cases the default gateway might not be able to reach the destination network, so you must also configure more specific static routes. For example, if the default gateway is outside, then the default route cannot direct traffic to any inside networks that are not directly connected to the security appliance.

In transparent firewall mode, for traffic that originates on the security appliance and is destined for a non-directly connected network, you need to configure either a default route or static routes so the security appliance knows out of which interface to send traffic. Traffic that originates on the security

appliance might include communications to a syslog server, Websense or N2H2 server, or AAA server. If you have servers that cannot all be reached through a single default route, then you must configure static routes.

The security appliance supports up to three equal cost routes on the same interface for load balancing.

This section includes the following topics:

- Configuring a Static Route, page 10-3
- Configuring a Default Static Route, page 10-4
- Configuring Static Route Tracking, page 10-5

For information about configuring IPv6 static and default routes, see the "Configuring IPv6 Default and Static Routes" section on page 13-5.

Configuring a Static Route

To add a static route, enter the following command:

```
hostname(config)# route if_name dest_ip mask gateway_ip [distance]
```

The *dest_ip* and *mask* is the IP address for the destination network and the *gateway_ip* is the address of the next-hop router.The addresses you specify for the static route are the addresses that are in the packet before entering the security appliance and performing NAT.

The *distance* is the administrative distance for the route. The default is 1 if you do not specify a value. Administrative distance is a parameter used to compare routes among different routing protocols. The default administrative distance for static routes is 1, giving it precedence over routes discovered by dynamic routing protocols but not directly connect routes. The default administrative distance for routes discovered by OSPF is 110. If a static route has the same administrative distance as a dynamic route, the static routes take precedence. Connected routes always take precedence over static or dynamically discovered routes.

Static routes remain in the routing table even if the specified gateway becomes unavailable. If the specified gateway becomes unavailable, you need to remove the static route from the routing table manually. However, static routes are removed from the routing table if the specified interface goes down. They are reinstated when the interface comes back up.

Note If you create a static route with an administrative distance greater than the administrative distance of the routing protocol running on the security appliance, then a route to the specified destination discovered by the routing protocol takes precedence over the static route. The static route is used only if the dynamically discovered route is removed from the routing table.

The following example creates a static route that sends all traffic destined for 10.1.1.0/24 to the router (10.1.2.45) connected to the inside interface:

```
hostname(config)# route inside 10.1.1.0 255.255.255.0 10.1.2.45 1
```

You can define up to three equal cost routes to the same destination per interface. ECMP is not supported across multiple interfaces. With ECMP, the traffic is not necessarily divided evenly between the routes; traffic is distributed among the specified gateways based on an algorithm that hashes the source and destination IP addresses.

The following example shows static routes that are equal cost routes that direct traffic to three different gateways on the outside interface. The security appliance distributes the traffic among the specified gateways.

```
hostname(config)# route outside 10.10.10.0 255.255.255.0 192.168.1.1
hostname(config)# route outside 10.10.10.0 255.255.255.0 192.168.1.2
hostname(config)# route outside 10.10.10.0 255.255.255.0 192.168.1.3
```

Configuring a Default Static Route

A default route identifies the gateway IP address to which the security appliance sends all IP packets for which it does not have a learned or static route. A default static route is simply a static route with 0.0.0.0/0 as the destination IP address. Routes that identify a specific destination take precedence over the default route.

Note In ASA software Versions 7.0 and later, if you have two default routes configured on different interfaces that have different metrics, the connection to the ASA firewall that is made from the higher metric interface fails, but connections to the ASA firewall from the lower metric interface succeed as expected. PIX software Version 6.3 supports connections from both the the higher and the lower metric interfaces.

You can define up to three equal cost default route entries per device. Defining more than one equal cost default route entry causes the traffic sent to the default route to be distributed among the specified gateways. When defining more than one default route, you must specify the same interface for each entry.

If you attempt to define more than three equal cost default routes, or if you attempt to define a default route with a different interface than a previously defined default route, you receive the message "ERROR: Cannot add route entry, possible conflict with existing routes."

You can define a separate default route for tunneled traffic along with the standard default route. When you create a default route with the **tunneled** option, all traffic from a tunnel terminating on the security appliance that cannot be routed using learned or static routes, is sent to this route. For traffic emerging from a tunnel, this route overrides over any other configured or learned default routes.

The following restrictions apply to default routes with the **tunneled** option:

- Do not enable unicast RPF (**ip verify reverse-path**) on the egress interface of tunneled route. Enabling uRPF on the egress interface of a tunneled route causes the session to fail.
- Do not enable TCP intercept on the egress interface of the tunneled route. Doing so causes the session to fail.
- Do not use the VoIP inspection engines (CTIQBE, H.323, GTP, MGCP, RTSP, SIP, SKINNY), the DNS inspect engine, or the DCE RPC inspection engine with tunneled routes. These inspection engines ignore the tunneled route.

You cannot define more than one default route with the **tunneled** option; ECMP for tunneled traffic is not supported.

To define the default route, enter the following command:

```
hostname(config)# route if_name 0.0.0.0 0.0.0.0 gateway_ip [distance | tunneled]
```

Tip You can enter 0 0 instead of 0.0.0.0 0.0.0.0 for the destination network address and mask, for example:

```
hostname(config)# route outside 0 0 192.168.1 1
```

The following example shows a security appliance configured with three equal cost default routes and a default route for tunneled traffic. Unencrypted traffic received by the security appliance for which there is no static or learned route is distributed among the gateways with the IP addresses 192.168.2.1, 192.168.2.2, 192.168.2.3. Encrypted traffic receive by the security appliance for which there is no static or learned route is passed to the gateway with the IP address 192.168.2.4.

```
hostname(config)# route outside 0 0 192.168.2.1
hostname(config)# route outside 0 0 192.168.2.2
hostname(config)# route outside 0 0 192.168.2.3
hostname(config)# route outside 0 0 192.168.2.4 tunneled
```

Configuring Static Route Tracking

One of the problems with static routes is that there is no inherent mechanism for determining if the route is up or down. They remain in the routing table even if the next hop gateway becomes unavailable. Static routes are only removed from the routing table if the associated interface on the security appliance goes down.

The static route tracking feature provides a method for tracking the availability of a static route and installing a backup route if the primary route should fail. This allows you to, for example, define a default route to an ISP gateway and a backup default route to a secondary ISP in case the primary ISP becomes unavailable.

The security appliance does this by associating a static route with a monitoring target that you define. It monitors the target using ICMP echo requests. If an echo reply is not received within a specified time period, the object is considered down and the associated route is removed from the routing table. A previously configured backup route is used in place of the removed route.

When selecting a monitoring target, you need to make sure it can respond to ICMP echo requests. The target can be any network object that you choose, but you should consider using:

- the ISP gateway (for dual ISP support) address
- the next hop gateway address (if you are concerned about the availability of the gateway)
- a server on the target network, such as a AAA server, that the security appliance needs to communicate with
- a persistent network object on the destination network (a desktop or notebook computer that may be shut down at night is not a good choice)

You can configure static route tracking for statically defined routes or default routes obtained through DHCP or PPPoE. You can only enable PPPoE clients on multiple interface with route tracking.

To configure static route tracking, perform the following steps:

Step 1 Configure the tracked object monitoring parameters:

a. Define the monitoring process:

```
hostname(config)# sla monitor sla_id
```

If you are configuring a new monitoring process, you are taken to SLA monitor configuration mode. If you are changing the monitoring parameters for an unscheduled monitoring process that already has a type defined, you are taken directly to the SLA protocol configuration mode.

b. Specify the monitoring protocol. If you are changing the monitoring parameters for an unscheduled monitoring process that already has a type defined, you are taken directly to SLA protocol configuration mode and cannot change this setting.

```
hostname(config-sla-monitor)# type echo protocol ipIcmpEcho target_ip interface
if_name
```

The *target_ip* is the IP address of the network object whose availability the tracking process monitors. While this object is available, the tracking process route is installed in the routing table. When this object becomes unavailable, the tracking process removed the route and the backup route is used in its place.

c. Schedule the monitoring process:

```
hostname(config)# sla monitor schedule sla_id [life {forever | seconds}] [start-time
{hh:mm[:ss] [month day | day month] | pending | now | after hh:mm:ss}] [ageout
seconds] [recurring]
```

Typically, you will use **sla monitor schedule** *sla_id* **life forever start-time now** for the monitoring schedule, and allow the monitoring configuration determine how often the testing occurs. However, you can schedule this monitoring process to begin in the future and to only occur at specified times.

Step 2 Associate a tracked static route with the SLA monitoring process by entering the following command:

```
hostname(config)# track track_id rtr sla_id reachability
```

The *track_id* is a tracking number you assign with this command. The *sla_id* is the ID number of the SLA process you defined in Step 1.

Step 3 Define the static route to be installed in the routing table while the tracked object is reachable using one of the following options:

- To track a static route, enter the following command:

```
hostname(config)# route if_name dest_ip mask gateway_ip [admin_distance] track
track_id
```

You cannot use the **tunneled** option with the **route** command with static route tracking.

- To track a default route obtained through DHCP, enter the following commands:

```
hostname(config)# interface phy_if
hostname(config-if)# dhcp client route track track_id
hostname(config-if)# ip addresss dhcp setroute
hostname(config-if)# exit
```

Note You must use the **setroute** argument with the **ip address dhcp** command to obtain the default route using DHCP.

- To track a default route obtained through PPPoE, enter the following commands:

```
hostname(config)# interface phy_if
hostname(config-if)# pppoe client route track track_id
hostname(config-if)# ip addresss pppoe setroute
hostname(config-if)# exit
```

Note You must use the **setroute** argument with the **ip address pppoe** command to obtain the default route using PPPoE.

Step 4 Define the backup route to use when the tracked object is unavailable using one of the following options. The administrative distance of the backup route must be greater than the administrative distance of the tracked route. If it is not, the backup route will be installed in the routing table instead of the tracked route.

- To use a static route, enter the following command:

```
hostname(config)# route if_name dest_ip mask gateway_ip [admin_distance]
```

The static route must have the same destination and mask as the tracked route. If you are tracking a default route obtained through DHCP or PPPoE, then the address and mask would be 0.0.0.0 0.0.0.0.

- To use a default route obtained through DHCP, enter the following commands:

```
hostname(config)# interface phy_if
hostname(config-if)# dhcp client route track track_id
hostname(config-if)# dhcp client route distance admin_distance
hostname(config-if)# ip addresss dhcp setroute
hostname(config-if)# exit
```

You must use the **setroute** argument with the **ip address dhcp** command to obtain the default route using DHCP. Make sure the administrative distance is greater than the administrative distance of the tracked route.

- To use a default route obtained through PPPoE, enter the following commands:

```
hostname(config)# interface phy_if
hostname(config-if)# pppoe client route track track_id
hostname(config-if)# pppoe client route distance admin_distance
hostname(config-if)# ip addresss pppoe setroute
hostname(config-if)# exit
```

You must use the **setroute** argument with the **ip address pppoe** command to obtain the default route using PPPoE. Make sure the administrative distance is greater than the administrative distance of the tracked route.

Defining Route Maps

Route maps are used when redistributing routes into an OSPF, RIP, or EIGRP routing process. They are also used when generating a default route into an OSPF routing process. A route map defines which of the routes from the specified routing protocol are allowed to be redistributed into the target routing process.

To define a route map, perform the following steps:

Step 1 To create a route map entry, enter the following command:

```
hostname(config)# route-map name {permit | deny} [sequence_number]
```

Route map entries are read in order. You can identify the order using the *sequence_number* option, or the security appliance uses the order in which you add the entries.

Step 2 Enter one or more **match** commands:

- To match any routes that have a destination network that matches a standard ACL, enter the following command:

```
hostname(config-route-map)# match ip address acl_id [acl_id] [...]
```

If you specify more than one ACL, then the route can match any of the ACLs.

- To match any routes that have a specified metric, enter the following command:

```
hostname(config-route-map)# match metric metric_value
```

The *metric_value* can be from 0 to 4294967295.

- To match any routes that have a next hop router address that matches a standard ACL, enter the following command:

```
hostname(config-route-map)# match ip next-hop acl_id [acl_id] [...]
```

If you specify more than one ACL, then the route can match any of the ACLs.

- To match any routes with the specified next hop interface, enter the following command:

```
hostname(config-route-map)# match interface if_name
```

If you specify more than one interface, then the route can match either interface.

- To match any routes that have been advertised by routers that match a standard ACL, enter the following command:

```
hostname(config-route-map)# match ip route-source acl_id [acl_id] [...]
```

If you specify more than one ACL, then the route can match any of the ACLs.

- To match the route type, enter the following command:

```
hostname(config-route-map)# match route-type {internal | external [type-1 | type-2]}
```

Step 3 Enter one or more **set** commands.

If a route matches the **match** commands, then the following **set** commands determine the action to perform on the route before redistributing it.

- To set the metric, enter the following command:

```
hostname(config-route-map)# set metric metric_value
```

The *metric_value* can be a value between 0 and 294967295

- To set the metric type, enter the following command:

```
hostname(config-route-map)# set metric-type {type-1 | type-2}
```

The following example shows how to redistribute routes with a hop count equal to 1 into OSPF. The security appliance redistributes these routes as external LSAs with a metric of 5, metric type of Type 1.

```
hostname(config)# route-map 1-to-2 permit
hostname(config-route-map)# match metric 1
hostname(config-route-map)# set metric 5
hostname(config-route-map)# set metric-type type-1
```

Configuring OSPF

This section describes how to configure OSPF. This section includes the following topics:

- OSPF Overview, page 10-9
- Enabling OSPF, page 10-10
- Redistributing Routes Into OSPF, page 10-10
- Configuring OSPF Interface Parameters, page 10-12

- Configuring OSPF Area Parameters, page 10-14
- Configuring OSPF NSSA, page 10-15
- Defining Static OSPF Neighbors, page 10-17
- Configuring Route Summarization Between OSPF Areas, page 10-16
- Configuring Route Summarization When Redistributing Routes into OSPF, page 10-16
- Generating a Default Route, page 10-17
- Configuring Route Calculation Timers, page 10-18
- Logging Neighbors Going Up or Down, page 10-18
- Displaying OSPF Update Packet Pacing, page 10-19
- Monitoring OSPF, page 10-19
- Restarting the OSPF Process, page 10-20

OSPF Overview

OSPF uses a link-state algorithm to build and calculate the shortest path to all known destinations. Each router in an OSPF area contains an identical link-state database, which is a list of each of the router usable interfaces and reachable neighbors.

The advantages of OSPF over RIP include the following:

- OSPF link-state database updates are sent less frequently than RIP updates, and the link-state database is updated instantly rather than gradually as stale information is timed out.
- Routing decisions are based on cost, which is an indication of the overhead required to send packets across a certain interface. The security appliance calculates the cost of an interface based on link bandwidth rather than the number of hops to the destination. The cost can be configured to specify preferred paths.

The disadvantage of shortest path first algorithms is that they require a lot of CPU cycles and memory.

The security appliance can run two processes of OSPF protocol simultaneously, on different sets of interfaces. You might want to run two processes if you have interfaces that use the same IP addresses (NAT allows these interfaces to coexist, but OSPF does not allow overlapping addresses). Or you might want to run one process on the inside, and another on the outside, and redistribute a subset of routes between the two processes. Similarly, you might need to segregate private addresses from public addresses.

You can redistribute routes into an OSPF routing process from another OSPF routing process, a RIP routing process, or from static and connected routes configured on OSPF-enabled interfaces.

The security appliance supports the following OSPF features:

- Support of intra-area, interarea, and external (Type I and Type II) routes.
- Support of a virtual link.
- OSPF LSA flooding.
- Authentication to OSPF packets (both password and MD5 authentication).
- Support for configuring the security appliance as a designated router or a designated backup router. The security appliance also can be set up as an ABR; however, the ability to configure the security appliance as an ASBR is limited to default information only (for example, injecting a default route).

- Support for stub areas and not-so-stubby-areas.
- Area boundary router type-3 LSA filtering.

Enabling OSPF

To enable OSPF, you need to create an OSPF routing process, specify the range of IP addresses associated with the routing process, then assign area IDs associated with that range of IP addresses.

To enable OSPF, perform the following steps:

Step 1 To create an OSPF routing process, enter the following command:

```
hostname(config)# router ospf process_id
```

This command enters the router configuration mode for this OSPF process.

The *process_id* is an internally used identifier for this routing process. It can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.

Step 2 To define the IP addresses on which OSPF runs and to define the area ID for that interface, enter the following command:

```
hostname(config-router)# network ip_address mask area area_id
```

The following example shows how to enable OSPF:

```
hostname(config)# router ospf 2
hostname(config-router)# network 10.0.0.0 255.0.0.0 area 0
```

Redistributing Routes Into OSPF

The security appliance can control the redistribution of routes between OSPF routing processes. The security appliance matches and changes routes according to settings in the **redistribute** command or by using a route map. See also the "Generating a Default Route" section on page 10-17 for another use for route maps.

To redistribute static, connected, RIP, or OSPF routes into an OSPF process, perform the following steps:

Step 1 (Optional) Create a route-map to further define which routes from the specified routing protocol are redistributed in to the OSPF routing process. See the "Defining Route Maps" section on page 10-7.

Step 2 If you have not already done so, enter the router configuration mode for the OSPF process you want to redistribute into by entering the following command:

```
hostname(config)# router ospf process_id
```

Step 3 Choose one of the following options to redistribute the selected route type into the RIP routing process.

- To redistribute connected routes into the OSPF routing process, enter the following command:

```
hostname(config-router): redistribute connected [metric metric-value]
[metric-type {type-1 | type-2}] [tag tag_value] [subnets] [route-map map_name]
```

- To redistribute static routes into the OSPF routing process, enter the following command:

  ```
  hostname(config-router): redistribute static [metric metric-value]
  [metric-type {type-1 | type-2}] [tag tag_value] [subnets] [route-map map_name]
  ```

- To redistribute routes from an OSPF routing process into the OSPF routing process, enter the following command:

  ```
  hostname(config-router): redistribute ospf pid [match {internal | external [1 | 2] |
  nssa-external [1 | 2]}] [metric metric-value] [metric-type {type-1 | type-2}]
  [tag tag_value] [subnets] [route-map map_name]
  ```

  You can either use the **match** options in this command to match and set route properties, or you can use a route map. The **tag** and **subnets** options do not have equivalents in the **route-map** command. If you use both a route map and **match** options in the **redistribute** command, then they must match

- To redistribute routes from a RIP routing process into the OSPF routing process, enter the following command:

  ```
  hostname(config-router): redistribute rip [metric metric-value]
  [metric-type {type-1 | type-2}] [tag tag_value] [subnets] [route-map map_name]
  ```

- To redistribute routes from an EIGRP routing process into the OSPF routing process, enter the following command:

  ```
  hostname(config-router): redistribute eigrp as-num [metric metric-value]
  [metric-type {type-1 | type-2}] [tag tag_value] [subnets] [route-map map_name]
  ```

The following example shows route redistribution from OSPF process 1 into OSPF process 2 by matching routes with a metric equal to 1. The security appliance redistributes these routes as external LSAs with a metric of 5, metric type of Type 1, and a tag equal to 1.

```
hostname(config)# route-map 1-to-2 permit
hostname(config-route-map)# match metric 1
hostname(config-route-map)# set metric 5
hostname(config-route-map)# set metric-type type-1
hostname(config-route-map)# set tag 1
hostname(config-route-map)# router ospf 2
hostname(config-router)# redistribute ospf 1 route-map 1-to-2
```

The following example shows the specified OSPF process routes being redistributed into OSPF process 109. The OSPF metric is remapped to 100.

```
hostname(config)# router ospf 109
hostname(config-router)# redistribute ospf 108 metric 100 subnets
```

The following example shows route redistribution where the link-state cost is specified as 5 and the metric type is set to external, indicating that it has lower priority than internal metrics.

```
hostname(config)# router ospf 1
hostname(config-router)# redistribute ospf 2 metric 5 metric-type external
```

Configuring OSPF Interface Parameters

You can alter some interface-specific OSPF parameters as necessary. You are not required to alter any of these parameters, but the following interface parameters must be consistent across all routers in an attached network: **ospf hello-interval**, **ospf dead-interval**, and **ospf authentication-key**. Be sure that if you configure any of these parameters, the configurations for all routers on your network have compatible values.

To configure OSPF interface parameters, perform the following steps:

Step 1 To enter the interface configuration mode, enter the following command:

```
hostname(config)# interface interface_name
```

Step 2 Enter any of the following commands:

- To specify the authentication type for an interface, enter the following command:

  ```
  hostname(config-interface)# ospf authentication [message-digest | null]
  ```

- To assign a password to be used by neighboring OSPF routers on a network segment that is using the OSPF simple password authentication, enter the following command:

  ```
  hostname(config-interface)# ospf authentication-key key
  ```

  The *key* can be any continuous string of characters up to 8 bytes in length.

  The password created by this command is used as a key that is inserted directly into the OSPF header when the security appliance software originates routing protocol packets. A separate password can be assigned to each network on a per-interface basis. All neighboring routers on the same network must have the same password to be able to exchange OSPF information.

- To explicitly specify the cost of sending a packet on an OSPF interface, enter the following command:

  ```
  hostname(config-interface)# ospf cost cost
  ```

  The *cost* is an integer from 1 to 65535.

- To set the number of seconds that a device must wait before it declares a neighbor OSPF router down because it has not received a hello packet, enter the following command:

  ```
  hostname(config-interface)# ospf dead-interval seconds
  ```

  The value must be the same for all nodes on the network.

- To specify the length of time between the hello packets that the security appliance sends on an OSPF interface, enter the following command:

  ```
  hostname(config-interface)# ospf hello-interval seconds
  ```

  The value must be the same for all nodes on the network.

- To enable OSPF MD5 authentication, enter the following command:

  ```
  hostname(config-interface)# ospf message-digest-key key_id md5 key
  ```

  Set the following values:

  - *key_id*—An identifier in the range from 1 to 255.
  - *key*—Alphanumeric password of up to 16 bytes.

Usually, one key per interface is used to generate authentication information when sending packets and to authenticate incoming packets. The same key identifier on the neighbor router must have the same key value.

We recommend that you not keep more than one key per interface. Every time you add a new key, you should remove the old key to prevent the local system from continuing to communicate with a hostile system that knows the old key. Removing the old key also reduces overhead during rollover.

- To set the priority to help determine the OSPF designated router for a network, enter the following command:

  ```
  hostname(config-interface)# ospf priority number_value
  ```

  The *number_value* is between 0 to 255.

- To specify the number of seconds between LSA retransmissions for adjacencies belonging to an OSPF interface, enter the following command:

  ```
  hostname(config-interface)# ospf retransmit-interval seconds
  ```

  The *seconds* must be greater than the expected round-trip delay between any two routers on the attached network. The range is from 1 to 65535 seconds. The default is 5 seconds.

- To set the estimated number of seconds required to send a link-state update packet on an OSPF interface, enter the following command:

  ```
  hostname(config-interface)# ospf transmit-delay seconds
  ```

  The *seconds* is from 1 to 65535 seconds. The default is 1 second.

- To specify the interface as a point-to-point, non-broadcast network, enter the following command:

  ```
  hostname(config-interface)# ospf network point-to-point non-broadcast
  ```

  When you designate an interface as point-to-point, non-broadcast, you must manually define the OSPF neighbor; dynamic neighbor discover is not possible. See Defining Static OSPF Neighbors, page 10-17, for more information. Additionally, you can only define one OSPF neighbor on that interface.

The following example shows how to configure the OSPF interfaces:

```
hostname(config)# router ospf 2
hostname(config-router)# network 2.0.0.0 255.0.0.0 area 0
hostname(config-router)# interface inside
hostname(config-interface)# ospf cost 20
hostname(config-interface)# ospf retransmit-interval 15
hostname(config-interface)# ospf transmit-delay 10
hostname(config-interface)# ospf priority 20
hostname(config-interface)# ospf hello-interval 10
hostname(config-interface)# ospf dead-interval 40
hostname(config-interface)# ospf authentication-key cisco
hostname(config-interface)# ospf message-digest-key 1 md5 cisco
hostname(config-interface)# ospf authentication message-digest
```

The following is sample output from the **show ospf** command:

```
hostname(config)# show ospf

Routing Process "ospf 2" with ID 20.1.89.2 and Domain ID 0.0.0.2
Supports only single TOS(TOS0) routes
Supports opaque LSA
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
```

```
Number of external LSA 5. Checksum Sum 0x 26da6
Number of opaque AS LSA 0. Checksum Sum 0x     0
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
External flood list length 0
    Area BACKBONE(0)
        Number of interfaces in this area is 1
        Area has no authentication
        SPF algorithm executed 2 times
        Area ranges are
        Number of LSA 5. Checksum Sum 0x 209a3
        Number of opaque link LSA 0. Checksum Sum 0x     0
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
```

Configuring OSPF Area Parameters

You can configure several area parameters. These area parameters (shown in the following task table) include setting authentication, defining stub areas, and assigning specific costs to the default summary route. Authentication provides password-based protection against unauthorized access to an area.

Stub areas are areas into which information on external routes is not sent. Instead, there is a default external route generated by the ABR, into the stub area for destinations outside the autonomous system. To take advantage of the OSPF stub area support, default routing must be used in the stub area. To further reduce the number of LSAs sent into a stub area, you can configure the **no-summary** keyword of the **area stub** command on the ABR to prevent it from sending summary link advertisement (LSA Type 3) into the stub area.

To specify area parameters for your network, perform the following steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
hostname(config)# router ospf process_id
```

Step 2 Enter any of the following commands:

- To enable authentication for an OSPF area, enter the following command:

  ```
  hostname(config-router)# area area-id authentication
  ```

- To enable MD5 authentication for an OSPF area, enter the following command:

  ```
  hostname(config-router)# area area-id authentication message-digest
  ```

- To define an area to be a stub area, enter the following command:

  ```
  hostname(config-router)# area area-id stub [no-summary]
  ```

- To assign a specific cost to the default summary route used for the stub area, enter the following command:

  ```
  hostname(config-router)# area area-id default-cost cost
  ```

  The *cost* is an integer from 1 to 65535. The default is 1.

The following example shows how to configure the OSPF area parameters:

```
hostname(config)# router ospf 2
hostname(config-router)# area 0 authentication
hostname(config-router)# area 0 authentication message-digest
hostname(config-router)# area 17 stub
hostname(config-router)# area 17 default-cost 20
```

Configuring OSPF NSSA

The OSPF implementation of an NSSA is similar to an OSPF stub area. NSSA does not flood type 5 external LSAs from the core into the area, but it can import autonomous system external routes in a limited way within the area.

NSSA importsType 7 autonomous system external routes within an NSSA area by redistribution. These Type 7 LSAs are translated into Type 5 LSAs by NSSA ABRs, which are flooded throughout the whole routing domain. Summarization and filtering are supported during the translation.

You can simplify administration if you are an ISP or a network administrator that must connect a central site using OSPF to a remote site that is using a different routing protocol using NSSA.

Before the implementation of NSSA, the connection between the corporate site border router and the remote router could not be run as an OSPF stub area because routes for the remote site could not be redistributed into the stub area, and two routing protocols needed to be maintained. A simple protocol such as RIP was usually run and handled the redistribution. With NSSA, you can extend OSPF to cover the remote connection by defining the area between the corporate router and the remote router as an NSSA.

To specify area parameters for your network as needed to configure OSPF NSSA, perform the following steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
hostname(config)# router ospf process_id
```

Step 2 Enter any of the following commands:

- To define an NSSA area, enter the following command:

  ```
  hostname(config-router)# area area-id nssa [no-redistribution]
  [default-information-originate]
  ```

- To summarize groups of addresses, enter the following command:

  ```
  hostname(config-router)# summary address ip_address mask [not-advertise] [tag tag]
  ```

  This command helps reduce the size of the routing table. Using this command for OSPF causes an OSPF ASBR to advertise one external route as an aggregate for all redistributed routes that are covered by the address.

  OSPF does not support **summary-address 0.0.0.0 0.0.0.0**.

  In the following example, the summary address 10.1.0.0 includes address 10.1.1.0, 10.1.2.0, 10.1.3.0, and so on. Only the address 10.1.0.0 is advertised in an external link-state advertisement:

  ```
  hostname(config-router)# summary-address 10.1.1.0 255.255.0.0
  ```

Before you use this feature, consider these guidelines:

- You can set a Type 7 default route that can be used to reach external destinations. When configured, the router generates a Type 7 default into the NSSA or the NSSA area boundary router.
- Every router within the same area must agree that the area is NSSA; otherwise, the routers will not be able to communicate.

Configuring Route Summarization Between OSPF Areas

Route summarization is the consolidation of advertised addresses. This feature causes a single summary route to be advertised to other areas by an area boundary router. In OSPF, an area boundary router advertises networks in one area into another area. If the network numbers in an area are assigned in a way such that they are contiguous, you can configure the area boundary router to advertise a summary route that covers all the individual networks within the area that fall into the specified range.

To define an address range for route summarization, perform the following steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
hostname(config)# router ospf process_id
```

Step 2 To set the address range, enter the following command:

```
hostname(config-router)# area area-id range ip-address mask [advertise | not-advertise]
```

The following example shows how to configure route summarization between OSPF areas:

```
hostname(config)# router ospf 1
hostname(config-router)# area 17 range 12.1.0.0 255.255.0.0
```

Configuring Route Summarization When Redistributing Routes into OSPF

When routes from other protocols are redistributed into OSPF, each route is advertised individually in an external LSA. However, you can configure the security appliance to advertise a single route for all the redistributed routes that are covered by a specified network address and mask. This configuration decreases the size of the OSPF link-state database.

To configure the software advertisement on one summary route for all redistributed routes covered by a network address and mask, perform the following steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
hostname(config)# router ospf process_id
```

Step 2 To set the summary address, enter the following command:

```
hostname(config-router)# summary-address ip_address mask [not-advertise] [tag tag]
```

Note OSPF does not support **summary-address 0.0.0.0 0.0.0.0**.

The following example shows how to configure route summarization. The summary address 10.1.0.0 includes address 10.1.1.0, 10.1.2.0, 10.1.3.0, and so on. Only the address 10.1.0.0 is advertised in an external link-state advertisement:

```
hostname(config)# router ospf 1
hostname(config-router)# summary-address 10.1.0.0 255.255.0.0
```

Defining Static OSPF Neighbors

You need to define static OSPF neighbors to advertise OSPF routes over a point-to-point, non-broadcast network. This lets you broadcast OSPF advertisements across an existing VPN connection without having to encapsulate the advertisements in a GRE tunnel.

To define a static OSPF neighbor, perform the following tasks:

Step 1 Create a static route to the OSPF neighbor. See the "Configuring Static and Default Routes" section on page 10-2 for more information about creating static routes.

Step 2 Define the OSPF neighbor by performing the following tasks:

a. Enter router configuration mode for the OSPF process. Enter the following command:

```
hostname(config)# router ospf pid
```

b. Define the OSPF neighbor by entering the following command:

```
hostname(config-router)# neighbor addr [interface if_name]
```

The *addr* argument is the IP address of the OSPF neighbor. The *if_name* is the interface used to communicate with the neighbor. If the OSPF neighbor is not on the same network as any of the directly-connected interfaces, you must specify the **interface**.

Generating a Default Route

You can force an autonomous system boundary router to generate a default route into an OSPF routing domain. Whenever you specifically configure redistribution of routes into an OSPF routing domain, the router automatically becomes an autonomous system boundary router. However, an autonomous system boundary router does not by default generate a default route into the OSPF routing domain.

To generate a default route, perform the following steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
hostname(config)# router ospf process_id
```

Step 2 To force the autonomous system boundary router to generate a default route, enter the following command:

```
hostname(config-router)# default-information originate [always] [metric metric-value]
[metric-type {1 | 2}] [route-map map-name]
```

The following example shows how to generate a default route:

```
hostname(config)# router ospf 2
hostname(config-router)# default-information originate always
```

Configuring Route Calculation Timers

You can configure the delay time between when OSPF receives a topology change and when it starts an SPF calculation. You also can configure the hold time between two consecutive SPF calculations.

To configure route calculation timers, perform the following steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
hostname(config)# router ospf process_id
```

Step 2 To configure the route calculation time, enter the following command:

```
hostname(config-router)# timers spf spf-delay spf-holdtime
```

The *spf-delay* is the delay time (in seconds) between when OSPF receives a topology change and when it starts an SPF calculation. It can be an integer from 0 to 65535. The default time is 5 seconds. A value of 0 means that there is no delay; that is, the SPF calculation is started immediately.

The *spf-holdtime* is the minimum time (in seconds) between two consecutive SPF calculations. It can be an integer from 0 to 65535. The default time is 10 seconds. A value of 0 means that there is no delay; that is, two SPF calculations can be done, one immediately after the other.

The following example shows how to configure route calculation timers:

```
hostname(config)# router ospf 1
hostname(config-router)# timers spf 10 120
```

Logging Neighbors Going Up or Down

By default, the system sends a system message when an OSPF neighbor goes up or down.

Configure this command if you want to know about OSPF neighbors going up or down without turning on the **debug ospf adjacency** command. The **log-adj-changes** router configuration command provides a higher level view of the peer relationship with less output. Configure **log-adj-changes detail** if you want to see messages for each state change.

To log neighbors going up or down, perform the following steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
hostname(config)# router ospf process_id
```

Step 2 To configure logging for neighbors going up or down, enter the following command:

```
hostname(config-router)# log-adj-changes [detail]
```

Note Logging must be enabled for the the neighbor up/down messages to be sent.

The following example shows how to log neighbors up/down messages:

```
hostname(config)# router ospf 1
hostname(config-router)# log-adj-changes detail
```

Displaying OSPF Update Packet Pacing

OSPF update packets are automatically paced so they are not sent less than 33 milliseconds apart. Without pacing, some update packets could get lost in situations where the link is slow, a neighbor could not receive the updates quickly enough, or the router could run out of buffer space. For example, without pacing packets might be dropped if either of the following topologies exist:

- A fast router is connected to a slower router over a point-to-point link.
- During flooding, several neighbors send updates to a single router at the same time.

Pacing is also used between resends to increase efficiency and minimize lost retransmissions. You also can display the LSAs waiting to be sent out an interface. The benefit of the pacing is that OSPF update and retransmission packets are sent more efficiently.

There are no configuration tasks for this feature; it occurs automatically.

To observe OSPF packet pacing by displaying a list of LSAs waiting to be flooded over a specified interface, enter the following command:

```
hostname# show ospf flood-list if_name
```

Monitoring OSPF

You can display specific statistics such as the contents of IP routing tables, caches, and databases. You can use the information provided to determine resource utilization and solve network problems. You can also display information about node reachability and discover the routing path that your device packets are taking through the network.

To display various OSPF routing statistics, perform one of the following tasks, as needed:

- To display general information about OSPF routing processes, enter the following command:

  ```
  hostname# show ospf [process-id [area-id]]
  ```

- To display the internal OSPF routing table entries to the ABR and ASBR, enter the following command:

  ```
  hostname# show ospf border-routers
  ```

- To display lists of information related to the OSPF database for a specific router, enter the following command:

  ```
  hostname# show ospf [process-id [area-id]] database
  ```

- To display a list of LSAs waiting to be flooded over an interface (to observe OSPF packet pacing), enter the following command:

  ```
  hostname# show ospf flood-list if-name
  ```

- To display OSPF-related interface information, enter the following command:

  ```
  hostname# show ospf interface [if_name]
  ```

- To display OSPF neighbor information on a per-interface basis, enter the following command:

  ```
  hostname# show ospf neighbor [interface-name] [neighbor-id] [detail]
  ```

- To display a list of all LSAs requested by a router, enter the following command:

  ```
  hostname# show ospf request-list neighbor if_name
  ```

- To display a list of all LSAs waiting to be resent, enter the following command:

  ```
  hostname# show ospf retransmission-list neighbor if_name
  ```

- To display a list of all summary address redistribution information configured under an OSPF process, enter the following command:

  ```
  hostname# show ospf [process-id] summary-address
  ```

- To display OSPF-related virtual links information, enter the following command:

  ```
  hostname# show ospf [process-id] virtual-links
  ```

Restarting the OSPF Process

To restart an OSPF process, clear redistribution, or counters, enter the following command:

```
hostname(config)# clear ospf pid {process | redistribution | counters
[neighbor [neighbor-interface] [neighbor-id]]}
```

Configuring RIP

Devices that support RIP send routing-update messages at regular intervals and when the network topology changes. These RIP packets contain information about the networks that the devices can reach, as well as the number of routers or gateways that a packet must travel through to reach the destination address. RIP generates more traffic than OSPF, but is easier to configure.

RIP has advantages over static routes because the initial configuration is simple, and you do not need to update the configuration when the topology changes. The disadvantage to RIP is that there is more network and processing overhead than static routing.

The security appliance supports RIP Version 1 and RIP Version 2.

This section describes how to configure RIP. This section includes the following topics:

- Enabling and Configuring RIP, page 10-21
- Redistributing Routes into the RIP Routing Process, page 10-22
- Configuring RIP Send/Receive Version on an Interface, page 10-23
- Enabling RIP Authentication, page 10-23
- Monitoring RIP, page 10-24

Enabling and Configuring RIP

You can only enable one RIP routing process on the security appliance. After you enable the RIP routing process, you must define the interfaces that will participate in that routing process using the **network** command. By default, the security appliance sends RIP Version 1 updates and accepts RIP Version 1 and Version 2 updates.

To enable and configure the RIP routing process, perform the following steps:

Step 1 Start the RIP routing process by entering the following command in global configuration mode:

```
hostname(config): router rip
```

You enter router configuration mode for the RIP routing process.

Step 2 Specify the interfaces that will participate in the RIP routing process. Enter the following command for each interface that will participate in the RIP routing process:

```
hostname(config-router): network network_address
```

If an interface belongs to a network defined by this command, the interface will participate in the RIP routing process. If an interface does not belong to a network defined by this command, it will not send or receive RIP updates.

Step 3 (Optional) Specify the version of RIP used by the security appliance by entering the following command:

```
hostname(config-router): version [1 | 2]
```

You can override this setting on a per-interface basis.

Step 4 (Optional) To generate a default route into RIP, enter the following command:

```
hostname(config-router): default-information originate
```

Step 5 (Optional) To specify an interface to operate in passive mode, enter the following command:

```
hostname(config-router): passive-interface [default | if_name]
```

Using the **default** keyword causes all interfaces to operate in passive mode. Specifying an interface name sets only that interface to passive RIP mode. In passive mode, RIP routing updates are accepted by but not sent out of the specified interface. You can enter this command for each interface you want to set to passive mode.

Step 6 (Optional) Disable automatic route summarization by entering the following command:

```
hostname(config-router): no auto-summarize
```

RIP Version 1 always uses automatic route summarization; you cannot disable it for RIP Version 1. RIP Version 2 uses route summarization by default; you can disable it using this command.

Step 7 (Optional) To filter the networks received in updates, perform the following steps:

a. Create a standard access list permitting the networks you want the RIP process to allow in the routing table and denying the networks you want the RIP process to discard.

b. Enter the following command to apply the filter. You can specify an interface to apply the filter to only those updates received by that interface.

```
hostname(config-router): distribute-list acl in [interface if_name]
```

You can enter this command for each interface you want to apply a filter to. If you do not specify an interface name, the filter is applied to all RIP updates.

Step 8 (Optional) To filter the networks sent in updates, perform the following steps:

a. Create a standard access list permitting the networks you want the RIP process to advertise and denying the networks you do not want the RIP process to advertise.

b. Enter the following command to apply the filter. You can specify an interface to apply the filter to only those updates sent by that interface.

```
hostname(config-router): distribute-list acl out [interface if_name]
```

You can enter this command for each interface you want to apply a filter to. If you do not specify an interface name, the filter is applied to all RIP updates.

Redistributing Routes into the RIP Routing Process

You can redistribute routes from the OSPF, EIGRP, static, and connected routing processes into the RIP routing process.

To redistribute a routes into the RIP routing process, perform the following steps:

Step 1 (Optional) Create a route-map to further define which routes from the specified routing protocol are redistributed in to the RIP routing process. See the "Defining Route Maps" section on page 10-7 for more information about creating a route map.

Step 2 Choose one of the following options to redistribute the selected route type into the RIP routing process.

- To redistribute connected routes into the RIP routing process, enter the following command:

```
hostname(config-router): redistribute connected [metric {metric_value | transparent}]
[route-map map_name]
```

- To redistribute static routes into the RIP routing process, enter the following command:

```
hostname(config-router): redistribute static [metric {metric_value | transparent}]
[route-map map_name]
```

- To redistribute routes from an OSPF routing process into the RIP routing process, enter the following command:

```
hostname(config-router): redistribute ospf pid [match {internal | external [1 | 2] |
nssa-external [1 | 2]}] [metric {metric_value | transparent}] [route-map map_name]
```

- To redistribute routes from an EIGRP routing process into the RIP routing process, enter the following command:

```
hostname(config-router): redistribute eigrp as-num [metric {metric_value |
transparent}] [route-map map_name]
```

Configuring RIP Send/Receive Version on an Interface

You can override the globally-set version of RIP the security appliance uses to send and receive RIP updates on a per-interface basis.

To configure the RIP send and receive version, perform the following steps:

Step 1 (Optional) To specify the version of RIP advertisements sent from an interface, perform the following steps:

a. Enter interface configuration mode for the interface you are configuring by entering the following command:

```
hostname(config)# interface phy_if
```

b. Specify the version of RIP to use when sending RIP updates out of the interface by entering the following command:

```
hostname(config-if)# rip send version {[1] [2]}
```

Step 2 (Optional) To specify the version of RIP advertisements permitted to be received by an interface, perform the following steps:

a. Enter interface configuration mode for the interface you are configuring by entering the following command:

```
hostname(config)# interface phy_if
```

b. Specify the version of RIP to allow when receiving RIP updates on the interface by entering the following command:

```
hostname(config-if)# rip receive version {[1] [2]}
```

RIP updates received on the interface that do not match the allowed version are dropped.

Enabling RIP Authentication

The security appliance supports RIP message authentication for RIP Version 2 messages.

To enable RIP message authentication, perform the following steps:

Step 1 Enter interface configuration mode for the interface you are configuring by entering the following command:

```
hostname(config)# interface phy_if
```

Step 2 (Optional) Set the authentication mode by entering the following command. By default, text authentication is used. MD5 authentication is recommended.

```
hostname(config-if)# rip authentication mode {text | md5}
```

Step 3 Enable authentication and configure the authentication key by entering the following command:

```
hostname(config-if)# rip authentication key key key_id key-id
```

Monitoring RIP

To display various RIP routing statistics, perform one of the following tasks, as needed:

- To display the contents of the RIP routing database, enter the following command:

  ```
  hostname# show rip database
  ```

- To display the RIP commands in the running configuration, enter the following command:

  ```
  hostname# show running-config router rip
  ```

Use the following **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC. Debugging output is assigned high priority in the CPU process and can render the system unusable. It is best to use **debug** commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased **debug** command processing overhead will affect system performance.

- To display RIP processing events, enter the following command:

  ```
  hostname# debug rip events
  ```

- To display RIP database events, enter the following command:

  ```
  hostname# debug rip database
  ```

Configuring EIGRP

This section describes the configuration and monitoring of EIGRP routing and includes the following topics:

- EIGRP Routing Overview, page 10-25
- Enabling and Configuring EIGRP Routing, page 10-26
- Enabling and Configuring EIGRP Stub Routing, page 10-27
- Enabling EIGRP Authentication, page 10-27
- Defining an EIGRP Neighbor, page 10-28
- Redistributing Routes Into EIGRP, page 10-29
- Configuring the EIGRP Hello Interval and Hold Time, page 10-30
- Disabling Automatic Route Summarization, page 10-30
- Configuring Summary Aggregate Addresses, page 10-31
- Disabling EIGRP Split Horizon, page 10-31

- Changing the Interface Delay Value, page 10-32
- Monitoring EIGRP, page 10-32
- Disabling Neighbor Change and Warning Message Logging, page 10-32

EIGRP Routing Overview

EIGRP is an enhanced version of IGRP developed by Cisco. Unlike IGRP and RIP, EIGRP does not send out periodic route updates. EIGRP updates are sent out only when the network topology changes.

Neighbor discovery is the process that the security appliance uses to dynamically learn of other routers on directly attached networks. EIGRP routers send out multicast hello packets to announce their presence on the network. When the security appliance receives a hello packet from a new neighbor, it sends its topology table to the neighbor with an initialization bit set. When the neighbor receives the topology update with the initialization bit set, the neighbor sends its topology table back to the security appliance.

The hello packets are sent out as multicast messages. No response is expected to a hello message. The exception to this is for statically defined neighbors. If you use the **neighbor** command to configure a neighbor, the hello messages sent to that neighbor are sent as unicast messages. Routing updates and acknowledgements are sent out as unicast messages.

Once this neighbor relationship is established, routing updates are not exchanged unless there is a change in the network topology. The neighbor relationship is maintained through the hello packets. Each hello packet received from a neighbor contains a hold time. This is the time in which the security appliance can expect to receive a hello packet from that neighbor. If the security appliance does not receive a hello packet from that neighbor within the hold time advertised by that neighbor, the security appliance considers that neighbor to be unavailable.

The EIGRP uses an algorithm called DUAL for route computations. DUAL saves all routes to a destination in the topology table, not just the least-cost route. The least-cost route is inserted into the routing table. The other routes remain in the topology table. If the main route fails, another route is chosen from the feasible successors. A successor is a neighboring router used for packet forwarding that has a least-cost path to a destination. The feasibility calculation guarantees that the path is not part of a routing loop.

If a feasible successor is not found in the topology table, a route recomputation must occur. During route recomputation, DUAL queries the EIGRP neighbors for a route, who in turn query their neighbors. Routers that do no have a feasible successor for the route return an unreachable message.

During route recomputation, DUAL marks the route as active. By default, the security appliance waits for three minutes to receive a response from its neighbors. If the security appliance does not receive a response from a neighbor, the route is marked as stuck-in-active. All routes in the topology table that point to the unresponsive neighbor as a feasibility successor are removed.



Note EIGRP neighbor relationships are not supported through the IPSec tunnel without a GRE tunnel.

Enabling and Configuring EIGRP Routing

You can only enable one EIGRP routing process on the security appliance.

To enable and configure EIGRP routing, perform the following tasks:

Step 1 Create the EIGRP routing process and enter router configuration mode for that process by entering the following command:

```
hostname(config)# router eigrp as-num
```

The *as-num* argument is the autonomous system number of the EIGRP routing process.

Step 2 To configure the interfaces and networks that participate in EIGRP routing, configure one or more **network** statements by entering the following command:

```
hostname(config-router)# network ip-addr [mask]
```

Directly-connected and static networks that fall within the defined network are advertised by the security appliance. Additionally, only interfaces with an IP address that fall within the defined network participate in the EIGRP routing process.

If you have an interface that you do not want to participate in EIGRP routing, but that is attached to a network that you want advertised, configure a **network** command that covers the network the interface is attached to, and use the **passive-interface** command to prevent that interface from sending or receiving EIGRP updates.

Step 3 (Optional) To prevent an interface from sending or receiving EIGRP routing message, enter the following command:

```
hostname(config-router)# passive-interface {default | if-name}
```

Using the **default** keyword disables EIGRP routing updates on all interfaces. Specifying an interface name, as defined by the **nameif** command, disables EIGRP routing updates on the specified interface. You can have multiple **passive-interface** commands in your EIGRP router configuration.

Step 4 (Optional) To control the sending or receiving of candidate default route information, enter the following command:

```
hostname(config-router)# no default-information {in | out}
```

Configuring **no default-information in** causes the candidate default route bit to be blocked on received routes. Configuring **no default-information out** disables the setting of th edefault route bit in advertised routes.

Step 5 (Optional) To filter networks sent in EIGRP routing updates, perform the following steps:

a. Create a standard access list that defines the routes you want to advertise.

b. Enter the following command to apply the filter. You can specify an interface to apply the filter to only those updates sent by that interface.

```
hostname(config-router): distribute-list acl out [interface if_name]
```

You can enter multiple **distribute-list** commands in your EIGRP router configuration.

Step 6 (Optional) To filter networks received in EIGRP routing updates, perform the following steps:

a. Create a standard access list that defines the routes you want to filter from received updates.

b. Enter the following command to apply the filter. You can specify an interface to apply the filter to only those updates received by that interface.

```
hostname(config-router): distribute-list acl in [interface if_name]
```

You can enter multiple **distribute-list** commands in your EIGRP router configuration.

Enabling and Configuring EIGRP Stub Routing

You can configure the security appliance as an EIGRP stub router. Stub routing decreases memory and processing requirements on the security appliance. As a stub router, the security appliance does not need to maintain a complete EIGRP routing table because it forwards all nonlocal traffic to a distribution router. Generally, the distribution router need not send anything more than a default route to the stub router.

Only specified routes are propagated from the stub router to the distribution router. As a stub router, the security appliance responds to all queries for summaries, connected routes, redistributed static routes, external routes, and internal routes with the message “inaccessible.” When the security appliance is configured as a stub, it sends a special peer information packet to all neighboring routers to report its status as a stub router. Any neighbor that receives a packet informing it of the stub status will not query the stub router for any routes, and a router that has a stub peer will not query that peer. The stub router depends on the distribution router to send the proper updates to all peers.

To enable and configure and EIGRP stub routing process, perform the following steps:

Step 1 Create the EIGRP routing process and enter router configuration mode for that process by entering the following command:

```
hostname(config)# router eigrp as-num
```

The *as-num* argument is the autonomous system number of the EIGRP routing process.

Step 2 Configure the interface connected to the distribution router to participate in EIGRP by entering the following command:

```
hostname(config-router)# network ip-addr [mask]
```

Step 3 Configure the stub routing process by entering the following command. You must specify which networks are advertised by the stub routing process to the distribution router. Static and connected networks are not automatically redistributed into the stub routing process.

```
hostname(config-router)# eigrp stub {receive-only | [connected] [redistributed] [static]
[summary]}
```

Enabling EIGRP Authentication

EIGRP route authentication provides MD5 authentication of routing updates from the EIGRP routing protocol. The MD5 keyed digest in each EIGRP packet prevents the introduction of unauthorized or false routing messages from unapproved sources.

EIGRP route authentication is configured on a per-interface basis. All EIGRP neighbors on interfaces configured for EIGRP message authentication must be configured with the same authentication mode and key for adjacencies to be established.

Before you can enable EIGRP route authentication, you must enable EIGRP.

To enable EIGRP authentication on an interface, perform the following steps:

Step 1 Enter interface configuration mode for the interface on which you are configuring EIGRP message authentication by entering the following command:

```
hostname(config)# interface phy_if
```

Step 2 Enable MD5 authentication of EIGRP packets by entering the following command:

```
hostname(config-if)# authentication mode eigrp as-num md5
```

The *as-num* argument is the autonomous system number of the EIGRP routing process configured on the security appliance. If EIGRP is not enabled or if you enter the wrong number, the security appliance returns the following error message:

```
% Asystem(100) specified does not exist
```

Step 3 Configure the key used by the MD5 algorithm by entering the following command:

```
hostname(config-if)# authentication key eigrp as-num key key-id key-id
```

The *as-num* argument is the autonomous system number of the EIGRP routing process configured on the security appliance. If EIGRP is not enabled or if you enter the wrong number, the security appliance returns the following error message:

```
% Asystem(100) specified does not exist
```

The *key* argument can contain up to 16 characters. The *key-id* argument is a number from 0 to 255.

Defining an EIGRP Neighbor

EIGRP hello packets are sent as multicast packets. If an EIGRP neighbor is located across a nonbroadcast network, such as a tunnel, you must manually define that neighbor. When you manually define an EIGRP neighbor, hello packets are sent to that neighbor as unicast messages.

To manually define an EIGRP neighbor, perform the following steps:

Step 1 Enter router configuration mode for the EIGRP routing process by entering the following command:

```
hostname(config)# router eigrp as-num
```

The *as-num* argument is the autonomous system number of the EIGRP routing process.

Step 2 Define the static neighbor by entering the following command:

```
hostname(config-router)# neighbor ip-addr interface if_name
```

The *ip-addr* argument is the IP address of the neighbor. The *if-name* argument is the name of the interface, as specified by the **nameif** command, through which that neighbor is available. You can define multiple neighbors for an EIGRP routing process.

Redistributing Routes Into EIGRP

You can redistribute routes discovered by RIP and OSPF into the EIGRP routing process. You can also redistribute static and connected routes into the EIGRP routing process. You do not need to redistribute connected routes if they fall within the range of a **network** statement in the EIGRP configuration.

To redistribute routes into the EIGRP routing process, perform the following steps:

Step 1 (Optional) Create a route-map to further define which routes from the specified routing protocol are redistributed in to the RIP routing process. See the "Defining Route Maps" section on page 10-7 for more information about creating a route map.

Step 2 Enter router configuration mode for the EIGRP routing process:

```
hostname(config)# router eigrp as-num
```

Step 3 (Optional) Specify the default metrics that should be applied to routes redistributed into the EIGRP routing process by entering the following command:

```
hostname(config-router)# default-metric bandwidth delay reliability loading mtu
```

If you do not specify a default-metric in the EIGRP router configuration, you must specify the metric values in each redistribute command. If you specify the EIGRP metrics in the redistribute command and have the default-metric command in the EIGRP router configuration, the metrics in the redistribute command are used. If you redistribute static or connected into EIGRP, specifying metric in **redistribute** command is not a requirement, though recommended.

Step 4 Choose one of the following options to redistribute the selected route type into the EIGRP routing process.

- To redistribute connected routes into the EIGRP routing process, enter the following command:

  ```
  hostname(config-router): redistribute connected [metric bandwidth delay reliability
  loading mtu] [route-map map_name]
  ```

- To redistribute static routes into the EIGRP routing process, enter the following command:

  ```
  hostname(config-router): redistribute static [metric bandwidth delay reliability
  loading mtu] [route-map map_name]
  ```

- To redistribute routes from an OSPF routing process into the EIGRP routing process, enter the following command:

  ```
  hostname(config-router): redistribute ospf pid [match {internal | external [1 | 2] |
  nssa-external [1 | 2]}] [metric bandwidth delay reliability loading mtu] [route-map
  map_name]
  ```

- To redistribute routes from a RIP routing process into the EIGRP routing process, enter the followin gcommand:

  ```
  hostname(config-router): redistribute rip [metric bandwidth delay reliability load mtu]
  [route-map map_name]
  ```

You must specify the EIGRP metric values in the **redistribute** command if you do not have a **default-metric** command in the EIGRP router configuration.

Configuring the EIGRP Hello Interval and Hold Time

The security appliance periodically sends hello packets to discover neighbors and to learn when neighbors become unreachable or inoperative. By default, hello packets are sent every 5 seconds.

The hello packet advertises the security appliance hold time. The hold time indicates to EIGRP neighbors the length of time the neighbor should consider the security appliance reachable. If the neighbor does not receive a hello packet within the advertised hold time, then the security appliance is considered unreachable. By default, the advertised hold time is 15 seconds (three times the hello interval).

Both the hello interval and the advertised hold time are configured on a per-interface basis. We recommend setting the hold time to be at minimum three times the hello interval.

To configure the hello interval and advertised hold time, perform the following steps:

Step 1 Enter interface configuration mode for the interface on which you are configuring hello interval or advertised hold time by entering the following command:

```
hostname(config)# interface phy_if
```

Step 2 To change the hello interval, enter the following command:

```
hostname(config)# hello-interval eigrp as-num seconds
```

Step 3 To change the hold time, enter the following command:

```
hostname(config)# hold-time eigrp as-num seconds
```

Disabling Automatic Route Summarization

Automatic route summarization is enabled by default. The EIGRP routing process summarizes on network number boundaries. This can cause routing problems if you have non-contiguous networks.

For example, if you have a router with the networks 192.168.1.0, 192.168.2.0, and 192.168.3.0 connected to it, and those networks all participate in EIGRP, the EIGRP routing process creates the summary address 192.168.0.0 for those routes. If an additional router is added to the network with the networks 192.168.10.0 and 192.168.11.0, and those networks participate in EIGRP, they will also be summarized as 192.168.0.0. To prevent the possibility of traffic being routed to the wrong location, you should disable automatic route summarization on the routers creating the conflicting summary addresses.

To disable automatic router summarization, enter the following command in router configuration mode for the EIGRP routing process:

```
hostname(config-router)# no auto-summary
```

Note Automatic summary addresses have an adminstrative distance of 5. You cannot configure this value.

Configuring Summary Aggregate Addresses

You can configure a summary addresses on a per-interface basis. You need to manually define summary addresses if you want to create summary addresses that do not occur at a network number boundary or if you want to use summary addresses on a security appliance with automatic route summarization disabled. If any more specific routes are in the routing table, EIGRP will advertise the summary address out the interface with a metric equal to the minimum of all more specific routes.

To create a summary address, perform the following steps:

Step 1 Enter interface configuration mode for the interface on which you are creating a summary address by entering the following command:

```
hostname(config)# interface phy_if
```

Step 2 Create the summary address by entering the following command:

```
hostname(config-if)# summary-address eigrp as-num address mask [distance]
```

By default, EIGRP summary addresses that you define have an administrative distance of 5. You can change this value by specifying the optional *distance* argument in the **summary-address** command.

Disabling EIGRP Split Horizon

Split horizon controls the sending of EIGRP update and query packets. When split horizon is enabled on an interface, update and query packets are not sent for destinations for which this interface is the next hop. Controlling update and query packets in this manner reduces the possibility of routing loops.

By default, split horizon is enabled on all interfaces.

Split horizon blocks route information from being advertised by a router out of any interface from which that information originated. This behavior usually optimizes communications among multiple routing devices, particularly when links are broken. However, with nonbroadcast networks, there may be situations where this behavior is not desired. For these situations, including networks in which you have EIGRP configured, you may want to disable split horizon.

If you disable split horizon on an interface, you must disable it for all routers and access servers on that interface.

To disable EIGRP split-horizon, perform the following steps:

Step 1 Enter interface configuration mode for the interface on which you are disabling split horizon by entering the following command:

```
hostname(config)# interface phy_if
```

Step 2 To disable split horizon, enter the following command:

```
hostname(config-if)# no split-horizon eigrp as-number
```

Changing the Interface Delay Value

The interface delay value is used in EIGRP distance calculations. You can modify this value on a per-interface basis.

To change the delay value, perform the following steps:

Step 1 Enter interface configuration mode for the interface on which you are changing the delay value used by EIGRP by entering the following command:

```
hostname(config)# interface phy_if
```

Step 2 To disable split horizon, enter the following command:

```
hostname(config-if)# delay value
```

The *value* entered is in tens of microseconds. So, to set the delay for 2000 microseconds, you would enter a *value* of 200.

Step 3 (Optional) To view the delay value assigned to an interface, use the **show interface** command.

Monitoring EIGRP

You can use the following commands to monitor the EIGRP routing process. For examples and descriptions of the command output, see the *Cisco Security Appliance Command Reference*.

- To display the EIGRP event log, enter the following command:

  ```
  hostname# show eigrp [as-number] events [{start end} | type]
  ```

- To display the interfaces participating in EIGRP routing, enter the following command:

  ```
  hostname# show eigrp [as-number] interfaces [if-name] [detail]
  ```

- To display the EIGRP neighbor table, enter the following command:

  ```
  hostname# show eigrp [as-number] neighbors [detail | static] [if-name]
  ```

- To display the EIGRP topology table, enter the following command:

  ```
  hostname# show eigrp [as-number] topology [ip-addr [mask] | active | all-links |
  pending | summary | zero-successors]
  ```

- To display EIGRP traffic statistics, enter the following command:

  ```
  hostname# show eigrp [as-number] traffic
  ```

Disabling Neighbor Change and Warning Message Logging

By default neighbor change, and neighbor warning messages are logged. You can disable the logging of neighbor change message and neighbor warning messages.

- To disable the logging of neighbor change messages, enter the following command in router configuration mode for the EIGRP routing process:

  ```
  hostname(config-router)# no eigrp log-neighbor-changes
  ```

- To disable the logging of neighbor warning messages, enter the following command in router configuration mode for the EIGRP routing process:

```
hostname(config-router)# no eigrp log-neighbor-warnings
```

The Routing Table

This section contains the following topics:

- Displaying the Routing Table, page 10-33
- How the Routing Table is Populated, page 10-33
- How Forwarding Decisions are Made, page 10-35

Displaying the Routing Table

To view the entries in the routing table, enter the following command:

```
hostname# show route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 10.86.194.1 to network 0.0.0.0

S    10.1.1.0 255.255.255.0 [3/0] via 10.86.194.1, outside
C    10.86.194.0 255.255.254.0 is directly connected, outside
S*   0.0.0.0 0.0.0.0 [1/0] via 10.86.194.1, outside
```

On the ASA 5505 adaptive security appliance, the following route is also shown. It is the internal loopback interface, which is used by the VPN hardware client feature for individual user authentication.

```
C 127.1.0.0 255.255.0.0 is directly connected, _internal_loopback
```

How the Routing Table is Populated

The security appliance routing table can be populated by statically defined routes, directly connected routes, and routes discovered by the RIP, EIGRP, and OSPF routing protocols. Because the security appliance can run multiple routing protocols in addition to having static and connected routed in the routing table, it is possible that the same route is discovered or entered in more than one manner. When two routes to the same destination are put into the routing table, the one that remains in the routing table is determined as follows:

- If the two routes have different network prefix lengths (network masks), then both routes are considered unique and are entered in to the routing table. The packet forwarding logic then determines which of the two to use.

  For example, if the RIP and OSPF processes discovered the following routes:

  - RIP: 192.168.32.0/24
  - OSPF: 192.168.32.0/19

  Even though OSPF routes have the better administrative distance, both routes are installed in the routing table because each of these routes has a different prefix length (subnet mask). They are considered different destinations and the packet forwarding logic determine which route to use.

- If the security appliance learns about multiple paths to the same destination from a single routing protocol, such as RIP, the route with the better metric (as determined by the routing protocol) is entered into the routing table.

  Metrics are values associated with specific routes, ranking them from most preferred to least preferred. The parameters used to determine the metrics differ for different routing protocols. The path with the lowest metric is selected as the optimal path and installed in the routing table. If there are multiple paths to the same destination with equal metrics, load balancing is done on these equal cost paths.

- If the security appliance learns about a destination from more than one routing protocol, the administrative distances of the routes are compared and the routes with lower administrative distance is entered into the routing table.

  You can change the administrative distances for routes discovered by or redistributed into a routing protocol. If two routes from two different routing protocols have the same administrative distance, then the route with the lower *default* administrative distance is entered into the routing table. In the case of EIGRP and OSPF routes, if the EIGRP route and the OSPF route have the same administrative distance, then the EIGRP route is chosen by default.

Administrative distance is a route parameter that the security appliance uses to select the best path when there are two or more different routes to the same destination from two different routing protocols. Because the routing protocols have metrics based on algorithms that are different from the other protocols, it is not always possible to determine the “best path” for two routes to the same destination that were generated by different routing protocols.

Each routing protocol is prioritized using an administrative distance value. Table 10-1 shows the default administrative distance values for the routing protocols supported by the security appliance.

Table 10-1 Default Administrative Distance for Supported Routing Protocols

Route Source	Default Administrative Distance
Connected interface	0
Static route	1
EIGRP Summary Route	5
Internal EIGRP	90
OSPF	110
RIP	120
EIGRP external route	170
Unknown	255

The smaller the administrative distance value, the more preference is given to the protocol. For example, if the security appliance receives a route to a certain network from both an OSPF routing process (default administrative distance - 110) and a RIP routing process (default administrative distance - 120), the security appliance chooses the OSPF route because OSPF has a higher preference. This means the router adds the OSPF version of the route to the routing table.

In the above example, if the source of the OSPF-derived route was lost (for example, due to a power shutdown), the security appliance would then use the RIP-derived route until the OSPF-derived route reappears.

The administrative distance is a local setting. For example, if you use the **distance-ospf** command to change the administrative distance of routes obtained through OSPF, that change would only affect the routing table for the security appliance the command was entered on. The administrative distance is not advertised in routing updates.

Administrative distance does not affect the routing process. The OSPF and RIP routing processes only advertise the routes that have been discovered by the routing process or redistributed into the routing process. For example, the RIP routing process advertises RIP routes, even if routes discovered by the OSPF routing process are used in the security appliance routing table.

Backup Routes

A backup route is registered when the initial attempt to install the route in the routing table fails because another route was installed instead. If the route that was installed in the routing table fails, the routing table maintenance process calls each routing protocol process that has registered a backup route and requests them to reinstall the route in the routing table. If there are multiple protocols with registered backup routes for the failed route, the preferred route is chosen based on administrative distance.

Because of this process, you can create "floating" static routes that are installed in the routing table when the route discovered by a dynamic routing protocol fails. A floating static route is simply a static route configured with a greater administrative distance than the dynamic routing protocols running on the security appliance. When the corresponding route discover by a dynamic routing process fails, the static route is installed in the routing table.

How Forwarding Decisions are Made

Forwarding decisions are made as follows:

- If the destination does not match an entry in the routing table, the packet is forwarded through the interface specified for the default route. If a default route has not been configured, the packet is discarded.
- If the destination matches a single entry in the routing table, the packet is forwarded through the interface associated with that route.
- If the destination matches more than one entry in the routing table, and the entries all have the same network prefix length, the packets for that destination are distributed among the interfaces associated with that route.
- If the destination matches more than one entry in the routing table, and the entries have different network prefix lengths, then the packet is forwarded out of the interface associated with the route that has the longer network prefix length.

For example, a packet destined for 192.168.32.1 arrives on an interface of a security appliance with the following routes in the routing table:

```
hostname# show route
         ....
         R   192.168.32.0/24 [120/4] via 10.1.1.2
         O   192.168.32.0/19 [110/229840] via 10.1.1.3
         ....
```

In this case, a packet destined to 192.168.32.1 is directed toward 10.1.1.2, because 192.168.32.1 falls within the 192.168.32.0/24 network. It also falls within the other route in the routing table, but the 192.168.32.0/24 has the longest prefix within the routing table (24 bits verses 19 bits). Longer prefixes are always preferred over shorter ones when forwarding a packet.

Dynamic Routing and Failover

Dynamic routes are not replicated to the standby unit or failover group in a failover configuration. Therefore, immediately after a failover occurs, some packets received by the security appliance may be dropped because of a lack of routing information or routed to a default static route while the routing table is repopulated by the configured dynamic routing protocols.



CHAPTER 11

Configuring DHCP, DDNS, and WCCP Services

This chapter describes how to configure the DHCP server, dynamic DNS (DDNS) update methods, and WCCP on the security appliance. DHCP provides network configuration parameters, such as IP addresses, to DHCP clients. The security appliance can provide a DHCP server or DHCP relay services to DHCP clients attached to security appliance interfaces. The DHCP server provides network configuration parameters directly to DHCP clients. DHCP relay passes DHCP requests received on one interface to an external DHCP server located behind a different interface.

DDNS update integrates DNS with DHCP. The two protocols are complementary: DHCP centralizes and automates IP address allocation; DDNS update automatically records the association between assigned addresses and hostnames at pre-defined intervals. DDNS allows frequently changing address-hostname associations to be updated frequently. Mobile hosts, for example, can then move freely on a network without user or administrator intervention. DDNS provides the necessary dynamic updating and synchronizing of the name to address and address to name mappings on the DNS server.

WCCP specifies interactions between one or more routers, Layer 3 switches, or security appliances and one or more web caches. The feature transparently redirects selected types of traffic to a group of web cache engines to optimize resource usage and lower response times.

Note The security appliance does not support QIP DHCP servers for use with DHCP Proxy.

This chapter includes the following sections:

- Configuring a DHCP Server, page 11-1
- Configuring DHCP Relay Services, page 11-5
- Configuring Dynamic DNS, page 11-6
- Configuring Web Cache Services Using WCCP, page 11-9

Configuring a DHCP Server

This section describes how to configure DHCP server provided by the security appliance. This section includes the following topics:

- Enabling the DHCP Server, page 11-2
- Configuring DHCP Options, page 11-3
- Using Cisco IP Phones with a DHCP Server, page 11-4

Enabling the DHCP Server

The security appliance can act as a DHCP server. DHCP is a protocol that supplies network settings to hosts including the host IP address, the default gateway, and a DNS server.

Note The security appliance DHCP server does not support BOOTP requests.

In multiple context mode, you cannot enable the DHCP server or DHCP relay on an interface that is used by more than one context.

You can configure a DHCP server on each interface of the security appliance. Each interface can have its own pool of addresses to draw from. However the other DHCP settings, such as DNS servers, domain name, options, ping timeout, and WINS servers, are configured globally and used by the DHCP server on all interfaces.

You cannot configure a DHCP client or DHCP Relay services on an interface on which the server is enabled. Additionally, DHCP clients must be directly connected to the interface on which the server is enabled.

When it receives a DHCP request, the security appliance sends a *discovery* message to the DHCP server. This message includes the IP address (within a subnetwork) configured with the **dhcp-network-scope** command in the group policy. If the server has an address pool that falls within that subnetwork, it sends the *offer* message with the pool information to the IP address—not to the source IP address of the discovery message.

For example, if the server has a pool of the range 209.165.200.225 to 209.165.200.254, mask 255.255.255.0, and the IP address specified by the **dhcp-network-scope** command is 209.165.200.1, the server sends that pool in the offer message to the security appliance.

To enable the DHCP server on a given security appliance interface, perform the following steps:

Step 1 Create a DHCP address pool. Enter the following command to define the address pool:

```
hostname(config)# dhcpd address ip_address-ip_address interface_name
```

The security appliance assigns a client one of the addresses from this pool to use for a given length of time. These addresses are the local, untranslated addresses for the directly connected network.

The address pool must be on the same subnet as the security appliance interface.

Step 2 (Optional) To specify the IP address(es) of the DNS server(s) the client will use, enter the following command:

```
hostname(config)# dhcpd dns dns1 [dns2]
```

You can specify up to two DNS servers.

Step 3 (Optional) To specify the IP address(es) of the WINS server(s) the client will use, enter the following command:

```
hostname(config)# dhcpd wins wins1 [wins2]
```

You can specify up to two WINS servers.

Step 4 (Optional) To change the lease length to be granted to the client, enter the following command:

```
hostname(config)# dhcpd lease lease_length
```

This lease equals the amount of time (in seconds) the client can use its allocated IP address before the lease expires. Enter a value between 300 to 1,048,575. The default value is 3600 seconds.

Step 5 (Optional) To configure the domain name the client uses, enter the following command:

```
hostname(config)# dhcpd domain domain_name
```

Step 6 (Optional) To configure the DHCP ping timeout value, enter the following command:

```
hostname(config)# dhcpd ping_timeout milliseconds
```

To avoid address conflicts, the security appliance sends two ICMP ping packets to an address before assigning that address to a DHCP client. This command specifies the timeout value for those packets.

Step 7 (Transparent Firewall Mode) Define a default gateway. To define the default gateway that is sent to DHCP clients, enter the following command.

```
hostname(config)# dhcpd option 3 ip gateway_ip
```

If you do not use the DHCP option 3 to define the default gateway, DHCP clients use the IP address of the management interface. The management interface does not route traffic.

Step 8 To enable the DHCP daemon within the security appliance to listen for DHCP client requests on the enabled interface, enter the following command:

```
hostname(config)# dhcpd enable interface_name
```

For example, to assign the range 10.0.1.101 to 10.0.1.110 to hosts connected to the inside interface, enter the following commands:

```
hostname(config)# dhcpd address 10.0.1.101-10.0.1.110 inside
hostname(config)# dhcpd dns 209.165.201.2 209.165.202.129
hostname(config)# dhcpd wins 209.165.201.5
hostname(config)# dhcpd lease 3000
hostname(config)# dhcpd domain example.com
hostname(config)# dhcpd enable inside
```

Configuring DHCP Options

You can configure the security appliance to send information for the DHCP options listed in RFC 2132. The DHCP options fall into one of three categories:

- Options that return an IP address.
- Options that return a text string.
- Options that return a hexadecimal value.

The security appliance supports all three categories of DHCP options. To configure a DHCP option, do one of the following:

- To configure a DHCP option that returns one or two IP addresses, enter the following command:

  ```
  hostname(config)# dhcpd option code ip addr_1 [addr_2]
  ```

- To configure a DHCP option that returns a text string, enter the following command:

  ```
  hostname(config)# dhcpd option code ascii text
  ```

- To configure a DHCP option that returns a hexadecimal value, enter the following command:

```
hostname(config)# dhcpd option code hex value
```

Note The security appliance does not verify that the option type and value that you provide match the expected type and value for the option code as defined in RFC 2132. For example, you can enter the **dhcpd option 46 ascii hello** command and the security appliance accepts the configuration although option 46 is defined in RFC 2132 as expecting a single-digit, hexadecimal value. For more information about the option codes and their associated types and expected values, refer to RFC 2132.

Table 11-1 shows the DHCP options that are not supported by the **dhcpd option** command.

Table 11-1 Unsupported DHCP Options

Option Code	Description
0	DHCPOPT_PAD
1	HCPOPT_SUBNET_MASK
12	DHCPOPT_HOST_NAME
50	DHCPOPT_REQUESTED_ADDRESS
51	DHCPOPT_LEASE_TIME
52	DHCPOPT_OPTION_OVERLOAD
53	DHCPOPT_MESSAGE_TYPE
54	DHCPOPT_SERVER_IDENTIFIER
58	DHCPOPT_RENEWAL_TIME
59	DHCPOPT_REBINDING_TIME
61	DHCPOPT_CLIENT_IDENTIFIER
67	DHCPOPT_BOOT_FILE_NAME
82	DHCPOPT_RELAY_INFORMATION
255	DHCPOPT_END

Specific options, DHCP option 3, 66, and 150, are used to configure Cisco IP Phones. See the "Using Cisco IP Phones with a DHCP Server" section on page 11-4 topic for more information about configuring those options.

Using Cisco IP Phones with a DHCP Server

Enterprises with small branch offices that implement a Cisco IP Telephony Voice over IP solution typically implement Cisco CallManager at a central office to control Cisco IP Phones at small branch offices. This implementation allows centralized call processing, reduces the equipment required, and eliminates the administration of additional Cisco CallManager and other servers at branch offices.

Cisco IP Phones download their configuration from a TFTP server. When a Cisco IP Phone starts, if it does not have both the IP address and TFTP server IP address preconfigured, it sends a request with option 150 or 66 to the DHCP server to obtain this information.

- DHCP option 150 provides the IP addresses of a list of TFTP servers.
- DHCP option 66 gives the IP address or the hostname of a single TFTP server.

Cisco IP Phones might also include DHCP option 3 in their requests, which sets the default route.

Cisco IP Phones might include both option 150 and 66 in a single request. In this case, the security appliance DHCP server provides values for both options in the response if they are configured on the security appliance.

You can configure the security appliance to send information for most options listed in RFC 2132. The following example shows the syntax for any option number, as well as the syntax for commonly-used options 66, 150, and 3:

- To provide information for DHCP requests that include an option number as specified in RFC-2132, enter the following command:

  ```
  hostname(config)# dhcpd option number value
  ```

- To provide the IP address or name of a TFTP server for option 66, enter the following command:

  ```
  hostname(config)# dhcpd option 66 ascii server_name
  ```

- To provide the IP address or names of one or two TFTP servers for option 150, enter the following command:

  ```
  hostname(config)# dhcpd option 150 ip server_ip1 [server_ip2]
  ```

  The *server_ip1* is the IP address or name of the primary TFTP server while *server_ip2* is the IP address or name of the secondary TFTP server. A maximum of two TFTP servers can be identified using option 150.

- To set the default route, enter the following command:

  ```
  hostname(config)# dhcpd option 3 ip router_ip1
  ```

Configuring DHCP Relay Services

A DHCP relay agent allows the security appliance to forward DHCP requests from clients to a router connected to a different interface.

The following restrictions apply to the use of the DHCP relay agent:

- The relay agent cannot be enabled if the DHCP server feature is also enabled.
- DHCP clients must be directly connected to the security appliance and cannot send requests through another relay agent or a router.
- For multiple context mode, you cannot enable DHCP relay on an interface that is used by more than one context.
- DHCP Relay services are not available in transparent firewall mode. A security appliance in transparent firewall mode only allows ARP traffic through; all other traffic requires an access list. To allow DHCP requests and replies through the security appliance in transparent mode, you need to configure two access lists, one that allows DCHP requests from the inside interface to the outside, and one that allows the replies from the server in the other direction.
- When DHCP relay is enabled and more than one DHCP relay server is defined, the security appliance forwards client requests to each defined DHCP relay server. Replies from the servers are also forwarded to the client until the client DHCP relay binding is removed. The binding is removed when the security appliance receives any of the following DHCP messages: ACK, NACK, or decline.

Note You cannot enable DHCP Relay on an interface running DHCP Proxy. You must Remove VPN DHCP configuration first or you will see an error message. This error happens if both DHCP relay and DHCP proxy are enabled. Ensure that either DHCP relay or DHCP proxy are enabled, but not both.

To enable DHCP relay, perform the following steps:

Step 1 To set the IP address of a DHCP server on a different interface from the DHCP client, enter the following command:

```
hostname(config)# dhcprelay server ip_address if_name
```

You can use this command up to 4 times to identify up to 4 servers.

Step 2 To enable DHCP relay on the interface connected to the clients, enter the following command:

```
hostname(config)# dhcprelay enable interface
```

Step 3 (Optional) To set the number of seconds allowed for relay address negotiation, enter the following command:

```
hostname(config)# dhcprelay timeout seconds
```

Step 4 (Optional) To change the first default router address in the packet sent from the DHCP server to the address of the security appliance interface, enter the following command:

```
hostname(config)# dhcprelay setroute interface_name
```

This action allows the client to set its default route to point to the security appliance even if the DHCP server specifies a different router.

If there is no default router option in the packet, the security appliance adds one containing the interface address.

The following example enables the security appliance to forward DHCP requests from clients connected to the inside interface to a DHCP server on the outside interface:

```
hostname(config)# dhcprelay server 201.168.200.4
hostname(config)# dhcprelay enable inside
hostname(config)# dhcprelay setroute inside
```

Configuring Dynamic DNS

This section describes examples for configuring the security appliance to support Dynamic DNS. DDNS update integrates DNS with DHCP. The two protocols are complementary—DHCP centralizes and automates IP address allocation, while dynamic DNS update automatically records the association between assigned addresses and hostnames. When you use DHCP and dynamic DNS update, this configures a host automatically for network access whenever it attaches to the IP network. You can locate and reach the host using its permanent, unique DNS hostname. Mobile hosts, for example, can move freely without user or administrator intervention.

DDNS provides address and domain name mappings so hosts can find each other even though their DHCP-assigned IP addresses change frequently. The DDNS name and address mappings are held on the DHCP server in two resource records: the A RR contains the name to IP address mapping while the PTR

RR maps addresses to names. Of the two methods for performing DDNS updates—the IETF standard defined by RFC 2136 and a generic HTTP method—the security appliance supports the IETF method in this release.

The two most common DDNS update configurations are:

- The DHCP client updates the A RR while the DHCP server updates PTR RR.
- The DHCP server updates both the A and PTR RRs.

In general, the DHCP server maintains DNS PTR RRs on behalf of clients. Clients may be configured to perform all desired DNS updates. The server may be configured to honor these updates or not. To update the PTR RR, the DHCP server must know the Fully Qualified Domain Name of the client. The client provides an FQDN to the server using a DHCP option called Client FQDN.

The following examples present these common scenarios:

- Example 1: Client Updates Both A and PTR RRs for Static IP Addresses, page 11-7
- Example 2: Client Updates Both A and PTR RRs; DHCP Server Honors Client Update Request; FQDN Provided Through Configuration, page 11-7
- Example 3: Client Includes FQDN Option Instructing Server Not to Update Either RR; Server Overrides Client and Updates Both RRs., page 11-8
- Example 4: Client Asks Server To Perform Both Updates; Server Configured to Update PTR RR Only; Honors Client Request and Updates Both A and PTR RR, page 11-9
- Example 5: Client Updates A RR; Server Updates PTR RR, page 11-9

Example 1: Client Updates Both A and PTR RRs for Static IP Addresses

The following example configures the client to request that it update both A and PTR resource records for static IP addresses. To configure this example, perform the following steps:

Step 1 To define a DDNS update method called ddns-2 that requests that the client update both the A and PTR RRs, enter the following commands:

```
hostname(config)# ddns update method ddns-2
hostname(DDNS-update-method)# ddns both
```

Step 2 To associate the method ddns-2 with the eth1 interface, enter the following commands:

```
hostname(DDNS-update-method)# interface eth1
hostname(config-if)# ddns update ddns-2
hostname(config-if)# ddns update hostname asa.example.com
```

Step 3 To configure a static IP address for eth1, enter the following commands:

```
hostname(config-if)# ip address 10.0.0.40 255.255.255.0
```

Example 2: Client Updates Both A and PTR RRs; DHCP Server Honors Client Update Request; FQDN Provided Through Configuration

The following example configures 1) the DHCP client to request that it update both the A and PTR RRs, and 2) the DHCP server to honor the requests. To configure this example, perform the following steps:

Step 1 To configure the DHCP client to request that the DHCP server perform no updates, enter the following command:

```
hostname(config)# dhcp-client update dns server none
```

Step 2 To create a DDNS update method named ddns-2 on the DHCP client that requests that the client perform both A and PTR updates, enter the following commands:

```
hostname(config)# ddns update method ddns-2
hostname(DDNS-update-method)# ddns both
```

Step 3 To associate the method named ddns-2 with the security appliance interface named Ethernet0, and enable DHCP on the interface, enter the following commands:

```
hostname(DDNS-update-method)# interface Ethernet0
hostname(if-config)# ddns update ddns-2
hostname(if-config)# ddns update hostname asa.example.com
hostname(if-config)# ip address dhcp
```

Step 4 To configure the DHCP server, enter the following command:

```
hostname(if-config)# dhcpd update dns
```

Example 3: Client Includes FQDN Option Instructing Server Not to Update Either RR; Server Overrides Client and Updates Both RRs.

The following example configures the DHCP client to include the FQDN option instructing the DHCP server not to update either the A or PTR updates. The example also configures the server to override the client request. As a result, the client backs off without performing any updates.

To configure this scenario, perform the following steps:

Step 1 To configure the update method named ddns-2 to request that it make both A and PTR RR updates, enter the following commands:

```
hostname(config)# ddns update method ddns-2
hostname(DDNS-update-method)# ddns both
```

Step 2 To assign the DDNS update method named ddns-2 on interface Ethernet0 and provide the client hostname (asa), enter the following commands:

```
hostname(DDNS-update-method)# interface Ethernet0
hostname(if-config)# ddns update ddns-2
hostname(if-config)# ddns update hostname asa.example.com
```

Step 3 To enable the DHCP client feature on the interface, enter the following commands:

```
hostname(if-config)# dhcp client update dns server none
hostname(if-config)# ip address dhcp
```

Step 4 To configure the DHCP server to override the client update requests, enter the following command:

```
hostname(if-config)# dhcpd update dns both override
```

Example 4: Client Asks Server To Perform Both Updates; Server Configured to Update PTR RR Only; Honors Client Request and Updates Both A and PTR RR

The following example configures the server to perform only PTR RR updates by default. However, the server honors the client request that it perform both A and PTR updates. The server also forms the FQDN by appending the domain name (example.com) to the hostname provided by the client (asa).

To configure this scenario, perform the following steps:

Step 1 To configure the DHCP client on interface Ethernet0, enter the following commands:

```
hostname(config)# interface Ethernet0
hostname(config-if)# dhcp client update dns both
hostname(config-if)# ddns update hostname asa
```

Step 2 To configure the DHCP server, enter the following commands:

```
hostname(config-if)# dhcpd update dns
hostname(config-if)# dhcpd domain example.com
```

Example 5: Client Updates A RR; Server Updates PTR RR

The following example configures the client to update the A resource record and the server to update the PTR records. Also, the client uses the domain name from the DHCP server to form the FQDN.

To configure this scenario, perform the following steps:

Step 1 To define the DDNS update method named ddns-2, enter the following commands:

```
hostname(config)# ddns update method ddns-2
hostname(DDNS-update-method)# ddns
```

Step 2 To configure the DHCP client for interface Ethernet0 and assign the update method to the interface, enter the following commands:

```
hostname(DDNS-update-method)# interface Ethernet0
hostname(config-if)# dhcp client update dns
hostname(config-if)# ddns update ddns-2
hostname(config-if)# ddns update hostname asa
```

Step 3 To configure the DHCP server, enter the following commands:

```
hostname(config-if)# dhcpd update dns
hostname(config-if)# dhcpd domain example.com
```

Configuring Web Cache Services Using WCCP

The purpose of web caching is to reduce latency and network traffic. Previously-accessed web pages are stored in a cache buffer, so if a user needs the page again, they can retrieve it from the cache instead of the web server.

WCCP specifies interactions between the security appliance and external web caches. The feature transparently redirects selected types of traffic to a group of web cache engines to optimize resource usage and lower response times. The security appliance only supports WCCP version 2.

Using a security appliance as an intermediary eliminates the need for a separate router to do the WCCP redirect because the security appliance takes care of redirecting requests to cache engines. When the security appliance knows when a packet needs redirection, it skips TCP state tracking, TCP sequence number randomization, and NAT on these traffic flows.

This section includes the following topics:

- WCCP Feature Support, page 11-10
- WCCP Interaction With Other Features, page 11-10
- Enabling WCCP Redirection, page 11-11

WCCP Feature Support

The following WCCPv2 features are supported with the security appliance:

- Redirection of multiple TCP/UDP port-destined traffic.
- Authentication for cache engines in a service group.

The following WCCPv2 features are not supported with the security appliance:

- Multiple routers in a service group is not supported. Multiple Cache Engines in a service group is still supported.
- Multicast WCCP is not supported.
- The Layer 2 redirect method is not supported; only GRE encapsulation is supported.
- WCCP source address spoofing.

WCCP Interaction With Other Features

In the security appliance implementation of WCCP, the following applies as to how the protocol interacts with other configurable features:

- An ingress access list entry always takes higher priority over WCCP. For example, if an access list does not permit a client to communicate with a server then traffic will not be redirected to a cache engine. Both ingress interface access lists and egress interface access lists will be applied.
- TCP intercept, authorization, URL filtering, inspect engines, and IPS features are not applied to a redirected flow of traffic.
- When a cache engine cannot service a request and packet is returned, or when a cache miss happens on a cache engine and it requests data from a web server, then the contents of the traffic flow will be subject to all the other configured features of the security appliance.
- In failover, WCCP redirect tables are not replicated to standby units. After a failover, packets will not be redirected until the tables are rebuilt. Sessions redirected prior to failover will likely be reset by the web server.

Enabling WCCP Redirection

There are two steps to configuring WCCP redirection on the security appliance. The first involves identifying the service to be redirected with the **wccp** command, and the second is defining on which interface the redirection occurs with the **wccp redirect** command. The **wccp** command can optionally also define which cache engines can participate in the service group, and what traffic should be redirected to the cache engine.

WCCP redirect is supported only on the ingress of an interface. The only topology that the security appliance supports is when client and cache engine are behind the same interface of the security appliance and the cache engine can directly communicate with the client without going through the security appliance.

The following configuration tasks assume you have already installed and configured the cache engines you wish to include in your network.

To configure WCCP redirection, perform the following steps:

Step 1 To enable a WCCP service group, enter the following command:

```
hostname(config)# wccp {web-cache | service_number} [redirect-list access_list]
[group-list access_list] [password password]
```

The standard service is **web-cache**, which intercepts TCP port 80 (HTTP) traffic and redirects that traffic to the cache engines, but you can identify a service number if desired between 0 and 254. For example, to transparently redirect native FTP traffic to a cache engine, use WCCP service 60. You can enter this command multiple times for each service group you want to enable.

The **redirect-list** *access_list* argument controls traffic redirected to this service group.

The **group-list** *access_list* argument determines which web cache IP addresses are allowed to participate in the service group.

The **password** *password* argument specifies MD5 authentication for messages received from the service group. Messages that are not accepted by the authentication are discarded.

Step 2 To enable WCCP redirection on an interface, enter the following command:

```
hostname(config)# wccp interface interface_name {web-cache | service_number} redirect in
```

The standard service is **web-cache**, which intercepts TCP port 80 (HTTP) traffic and redirects that traffic to the cache engines, but you can identify a service number if desired between 0 and 254. For example, to transparently redirect native FTP traffic to a cache engine, use WCCP service 60. You can enter this command multiple times for each service group you want to participate in.

For example, to enable the standard **web-cache** service and redirect HTTP traffic that enters the inside interface to a web cache, enter the following commands:

```
hostname(config)# wccp web-cache
hostname(config)# wccp interface inside web-cache redirect in
```



CHAPTER 12

Configuring Multicast Routing

This chapter describes how to configure multicast routing. This chapter includes the following topics:

- Multicast Routing Overview, page 12-1
- Enabling Multicast Routing, page 12-2
- Configuring IGMP Features, page 12-2
- Configuring Stub Multicast Routing, page 12-5
- Configuring a Static Multicast Route, page 12-6
- Configuring PIM Features, page 12-6
- For More Information about Multicast Routing, page 12-10

Multicast Routing Overview

The security appliance supports both stub multicast routing and PIM multicast routing. However, you cannot configure both concurrently on a single security appliance.

Note Only the UDP transport layer is supported for multicast routing.

Stub multicast routing provides dynamic host registration and facilitates multicast routing. When configured for stub multicast routing, the security appliance acts as an IGMP proxy agent. Instead of fully participating in multicast routing, the security appliance forwards IGMP messages to an upstream multicast router, which sets up delivery of the multicast data. When configured for stub multicast routing, the security appliance cannot be configured for PIM.

The security appliance supports both PIM-SM and bi-directional PIM. PIM-SM is a multicast routing protocol that uses the underlying unicast routing information base or a separate multicast-capable routing information base. It builds unidirectional shared trees rooted at a single Rendezvous Point per multicast group and optionally creates shortest-path trees per multicast source.

Bi-directional PIM is a variant of PIM-SM that builds bi-directional shared trees connecting multicast sources and receivers. Bi-directional trees are built using a DF election process operating on each link of the multicast topology. With the assistance of the DF, multicast data is forwarded from sources to the Rendezvous Point, and therefore along the shared tree to receivers, without requiring source-specific state. The DF election takes place during Rendezvous Point discovery and provides a default route to the Rendezvous Point.

Note If the security appliance is the PIM RP, use the untranslated outside address of the security appliance as the RP address.

Enabling Multicast Routing

Enabling multicast routing lets the security appliance forward multicast packets. Enabling multicast routing automatically enables PIM and IGMP on all interfaces. To enable multicast routing, enter the following command:

```
hostname(config)# multicast-routing
```

The number of entries in the multicast routing tables are limited by the amount of RAM on the system. Table 12-1 lists the maximum number of entries for specific multicast tables based on the amount of RAM on the security appliance. Once these limits are reached, any new entries are discarded.

Table 12-1 Entry Limits for Multicast Tables

Table	16 MB	128 MB	128+ MB
MFIB	1000	3000	5000
IGMP Groups	1000	3000	5000
PIM Routes	3000	7000	12000

Configuring IGMP Features

IP hosts use IGMP to report their group memberships to directly connected multicast routers. IGMP uses group addresses (Class D IP address) as group identifiers. Host group address can be in the range 224.0.0.0 to 239.255.255.255. The address 224.0.0.0 is never assigned to any group. The address 224.0.0.1 is assigned to all systems on a subnet. The address 224.0.0.2 is assigned to all routers on a subnet.

When you enable multicast routing on the security appliance, IGMP Version 2 is automatically enabled on all interfaces.

Note Only the **no igmp** command appears in the interface configuration when you use the **show run** command. If the **multicast-routing** command appears in the device configuration, then IGMP is automatically enabled on all interfaces.

This section describes how to configure optional IGMP setting on a per-interface basis. This section includes the following topics:

- Disabling IGMP on an Interface, page 12-3
- Configuring Group Membership, page 12-3
- Configuring a Statically Joined Group, page 12-3
- Controlling Access to Multicast Groups, page 12-3
- Limiting the Number of IGMP States on an Interface, page 12-4
- Modifying the Query Interval and Query Timeout, page 12-4

- Changing the Query Response Time, page 12-5
- Changing the IGMP Version, page 12-5

Disabling IGMP on an Interface

You can disable IGMP on specific interfaces. This is useful if you know that you do not have any multicast hosts on a specific interface and you want to prevent the security appliance from sending host query messages on that interface.

To disable IGMP on an interface, enter the following command:

```
hostname(config-if)# no igmp
```

To reenable IGMP on an interface, enter the following command:

```
hostname(config-if)# igmp
```

Note Only the **no igmp** command appears in the interface configuration.

Configuring Group Membership

You can configure the security appliance to be a member of a multicast group. Configuring the security appliance to join a multicast group causes upstream routers to maintain multicast routing table information for that group and keep the paths for that group active.

To have the security appliance join a multicast group, enter the following command:

```
hostname(config-if)# igmp join-group group-address
```

Configuring a Statically Joined Group

Sometimes a group member cannot report its membership in the group, or there may be no members of a group on the network segment, but you still want multicast traffic for that group to be sent to that network segment. You can have multicast traffic for that group sent to the segment in one of two ways:

- Using the **igmp join-group** command (see Configuring Group Membership, page 12-3). This causes the security appliance to accept and to forward the multicast packets.
- Using the **igmp static-group** command. The security appliance does not accept the multicast packets but rather forwards them to the specified interface.

To configure a statically joined multicast group on an interface, enter the following command:

```
hostname(config-if)# igmp static-group group-address
```

Controlling Access to Multicast Groups

To control the multicast groups that hosts on the security appliance interface can join, perform the following steps:

Step 1 Create an access list for the multicast traffic. You can create more than one entry for a single access list. You can use extended or standard access lists.

- To create a standard access list, enter the following command:

  ```
  hostname(config)# access-list name standard [permit | deny] ip_addr mask
  ```

  The *ip_addr* argument is the IP address of the multicast group being permitted or denied.

- To create an extended access list, enter the following command:

  ```
  hostname(config)# access-list name extended [permit | deny] protocol src_ip_addr
  src_mask dst_ip_addr dst_mask
  ```

  The *dst_ip_addr* argument is the IP address of the multicast group being permitted or denied.

Step 2 Apply the access list to an interface by entering the following command:

```
hostname(config-if)# igmp access-group acl
```

The *acl* argument is the name of a standard or extended IP access list.

Limiting the Number of IGMP States on an Interface

You can limit the number of IGMP states resulting from IGMP membership reports on a per-interface basis. Membership reports exceeding the configured limits are not entered in the IGMP cache and traffic for the excess membership reports is not forwarded.

To limit the number of IGMP states on an interface, enter the following command:

```
hostname(config-if)# igmp limit number
```

Valid values range from 0 to 500, with 500 being the default value. Setting this value to 0 prevents learned groups from being added, but manually defined memberships (using the **igmp join-group** and **igmp static-group** commands) are still permitted. The **no** form of this command restores the default value.

Modifying the Query Interval and Query Timeout

The security appliance sends query messages to discover which multicast groups have members on the networks attached to the interfaces. Members respond with IGMP report messages indicating that they want to receive multicast packets for specific groups. Query messages are addressed to the all-systems multicast group, which has an address of 224.0.0.1, with a time-to-live value of 1.

These messages are sent periodically to refresh the membership information stored on the security appliance. If the security appliance discovers that there are no local members of a multicast group still attached to an interface, it stops forwarding multicast packet for that group to the attached network and it sends a prune message back to the source of the packets.

By default, the PIM designated router on the subnet is responsible for sending the query messages. By default, they are sent once every 125 seconds. To change this interval, enter the following command:

```
hostname(config-if)# igmp query-interval seconds
```

If the security appliance does not hear a query message on an interface for the specified timeout value (by default, 255 seconds), then the security appliance becomes the designated router and starts sending the query messages. To change this timeout value, enter the following command:

```
hostname(config-if)# igmp query-timeout seconds
```



Note The **igmp query-timeout** and **igmp query-interval** commands require IGMP Version 2.

Changing the Query Response Time

By default, the maximum query response time advertised in IGMP queries is 10 seconds. If the security appliance does not receive a response to a host query within this amount of time, it deletes the group.

To change the maximum query response time, enter the following command:

```
hostname(config-if)# igmp query-max-response-time seconds
```

Changing the IGMP Version

By default, the security appliance runs IGMP Version 2, which enables several additional features such as the **igmp query-timeout** and **igmp query-interval** commands.

All multicast routers on a subnet must support the same version of IGMP. The security appliance does not automatically detect version 1 routers and switch to version 1. However, a mix of IGMP Version 1 and 2 hosts on the subnet works; the security appliance running IGMP Version 2 works correctly when IGMP Version 1 hosts are present.

To control which version of IGMP is running on an interface, enter the following command:

```
hostname(config-if)# igmp version {1 | 2}
```

Configuring Stub Multicast Routing

A security appliance acting as the gateway to the stub area does not need to participate in PIM. Instead, you can configure it to act as an IGMP proxy agent and forward IGMP messages from hosts connected on one interface to an upstream multicast router on another. To configure the security appliance as an IGMP proxy agent, forward the host join and leave messages from the stub area interface to an upstream interface.

To forward the host join and leave messages, enter the following command from the interface attached to the stub area:

```
hostname(config-if)# igmp forward interface if_name
```



Note Stub Multicast Routing and PIM are not supported concurrently.

Configuring a Static Multicast Route

When using PIM, the security appliance expects to receive packets on the same interface where it sends unicast packets back to the source. In some cases, such as bypassing a route that does not support multicast routing, you may want unicast packets to take one path and multicast packets to take another.

Static multicast routes are not advertised or redistributed.

To configure a static multicast route for PIM, enter the following command:

```
hostname(config)# mroute src_ip src_mask {input_if_name | rpf_neighbor} [distance]
```

To configure a static multicast route for a stub area, enter the following command:

```
hostname(config)# mroute src_ip src_mask input_if_name [dense output_if_name] [distance]
```

Note The **dense** *output_if_name* keyword and argument pair is only supported for stub multicast routing.

Configuring PIM Features

Routers use PIM to maintain forwarding tables for forwarding multicast diagrams. When you enable multicast routing on the security appliance, PIM and IGMP are automatically enabled on all interfaces.

Note PIM is not supported with PAT. The PIM protocol does not use ports and PAT only works with protocols that use ports.

This section describes how to configure optional PIM settings. This section includes the following topics:

- Disabling PIM on an Interface, page 12-6
- Configuring a Static Rendezvous Point Address, page 12-7
- Configuring the Designated Router Priority, page 12-7
- Filtering PIM Register Messages, page 12-7
- Configuring PIM Message Intervals, page 12-8
- Configuring a Multicast Boundary, page 12-8
- Filtering PIM Neighbors, page 12-8
- Supporting Mixed Bidirectional/Sparse-Mode PIM Networks, page 12-9

Disabling PIM on an Interface

You can disable PIM on specific interfaces. To disable PIM on an interface, enter the following command:

```
hostname(config-if)# no pim
```

To reenable PIM on an interface, enter the following command:

```
hostname(config-if)# pim
```

Note Only the **no pim** command appears in the interface configuration.

Configuring a Static Rendezvous Point Address

All routers within a common PIM sparse mode or bidir domain require knowledge of the PIM RP address. The address is statically configured using the **pim rp-address** command.

Note The security appliance does not support Auto-RP or PIM BSR; you must use the **pim rp-address** command to specify the RP address.

You can configure the security appliance to serve as RP to more than one group. The group range specified in the access list determines the PIM RP group mapping. If an access list is not specified, then the RP for the group is applied to the entire multicast group range (224.0.0.0/4).

To configure the address of the PIM PR, enter the following command:

```
hostname(config)# pim rp-address ip_address [acl] [bidir]
```

The *ip_address* argument is the unicast IP address of the router to be a PIM RP. The *acl* argument is the name or number of a standard access list that defines which multicast groups the RP should be used with. Do not use a host ACL with this command. Excluding the **bidir** keyword causes the groups to operate in PIM sparse mode.

Note The security appliance always advertises the bidir capability in the PIM hello messages regardless of the actual bidir configuration.

Configuring the Designated Router Priority

The DR is responsible for sending PIM register, join, and prune messaged to the RP. When there is more than one multicast router on a network segment, there is an election process to select the DR based on DR priority. If multiple devices have the same DR priority, then the device with the highest IP address becomes the DR.

By default, the security appliance has a DR priority of 1. You can change this value by entering the following command:

```
hostname(config-if)# pim dr-priority num
```

The *num* argument can be any number from 1 to 4294967294.

Filtering PIM Register Messages

You can configure the security appliance to filter PIM register messages. To filter PIM register messages, enter the following command:

```
hostname(config)# pim accept-register {list acl | route-map map-name}
```

Configuring PIM Message Intervals

Router query messages are used to elect the PIM DR. The PIM DR is responsible for sending router query messages. By default, router query messages are sent every 30 seconds. You can change this value by entering the following command:

```
hostname(config-if)# pim hello-interval seconds
```

Valid values for the *seconds* argument range from 1 to 3600 seconds.

Every 60 seconds, the security appliance sends PIM join/prune messages. To change this value, enter the following command:

```
hostname(config-if)# pim join-prune-interval seconds
```

Valid values for the *seconds* argument range from 10 to 600 seconds.

Configuring a Multicast Boundary

Address scoping defines domain boundaries so that domains with RPs that have the same IP address do not leak into each other. Scoping is performed on the subnet boundaries within large domains and on the boundaries between the domain and the Internet.

You can set up an administratively scoped boundary on an interface for multicast group addresses using the **multicast boundary** command. IANA has designated the multicast address range 239.0.0.0 to 239.255.255.255 as the administratively scoped addresses. This range of addresses can be reused in domains administered by different organizations. They would be considered local, not globally unique.

To configure a multicast boundary, enter the following command:

```
hostname(config-if)# multicast boundary acl [filter-autorp]
```

A standard ACL defines the range of addresses affected. When a boundary is set up, no multicast data packets are allowed to flow across the boundary from either direction. The boundary allows the same multicast group address to be reused in different administrative domains.

You can configure the **filter-autorp** keyword to examine and filter Auto-RP discovery and announcement messages at the administratively scoped boundary. Any Auto-RP group range announcements from the Auto-RP packets that are denied by the boundary access control list (ACL) are removed. An Auto-RP group range announcement is permitted and passed by the boundary only if all addresses in the Auto-RP group range are permitted by the boundary ACL. If any address is not permitted, the entire group range is filtered and removed from the Auto-RP message before the Auto-RP message is forwarded.

Filtering PIM Neighbors

You can define the routers that can become PIM neighbors with the **pim neighbor-filter** command. By filtering the routers that can become PIM neighbors, you can:

- Prevent unauthorized routers from becoming PIM neighbors.
- Prevent attached stub routers from participating in PIM.

To define the neighbors that can become a PIM neighbor, perform the following steps:

Step 1 Use the **access-list** command to define a standard access list defines the routers you want to participate in PIM.

For example the following access list, when used with the **pim neighbor-filter** command, prevents the 10.1.1.1 router from becoming a PIM neighbor:

```
hostname(config)# access-list pim_nbr deny 10.1.1.1 255.255.255.255
```

Step 2 Use the **pim neighbor-filter** command on an interface to filter the neighbor routers.

For example, the following commands prevent the 10.1.1.1 router from becoming a PIM neighbor on interface GigabitEthernet0/3:

```
hostname(config)# interface GigabitEthernet0/3
hostname(config-if)# pim neighbor-filter pim_nbr
```

Supporting Mixed Bidirectional/Sparse-Mode PIM Networks

Bidirectional PIM allows multicast routers to keep reduced state information. All of the multicast routers in a segment must be bidirectionally enabled in order for bidir to elect a DF.

The **pim bidir-neighbor-filter** command enables the transition from a sparse-mode-only network to a bidir network by letting you specify the routers that should participate in DF election while still allowing all routers to participate in the sparse-mode domain. The bidir-enabled routers can elect a DF from among themselves, even when there are non-bidir routers on the segment. Multicast boundaries on the non-bidir routers prevent PIM messages and data from the bidir groups from leaking in or out of the bidir subset cloud.

When the **pim bidir-neighbor-filter** command is enabled, the routers that are permitted by the ACL are considered to be bidir-capable. Therefore:

- If a permitted neighbor does not support bidir, the DF election does not occur.
- If a denied neighbor supports bidir, then DF election does not occur.
- If a denied neighbor des not support bidir, the DF election occurs.

To control which neighbors can participate in the DF election, perform the following steps:

Step 1 Use the **access-list** command to define a standard access list that permits the routers you want to participate in the DF election and denies all others.

For example, the following access list permits the routers at 10.1.1.1 and 10.2.2.2 to participate in the DF election and denies all others:

```
hostname(config)# access-list pim_bidir permit 10.1.1.1 255.255.255.255
hostname(config)# access-list pim_bidir permit 10.1.1.2 255.255.255.255
hostname(config)# access-list pim_bidir deny any
```

Step 2 Enable the **pim bidir-neighbor-filter** command on an interface.

The following example applies the access list created previous step to the interface GigabitEthernet0/3.

```
hostname(config)# interface GigabitEthernet0/3
hostname(config-if)# pim bidir-neighbor-filter pim_bidir
```

For More Information about Multicast Routing

The following RFCs from the IETF provide technical details about the IGMP and multicast routing standards used for implementing the SMR feature:

- RFC 2236 IGMPv2
- RFC 2362 PIM-SM
- RFC 2588 IP Multicast and Firewalls
- RFC 2113 IP Router Alert Option
- IETF draft-ietf-idmr-igmp-proxy-01.txt



CHAPTER 13

Configuring IPv6

This chapter describes how to enable and configure IPv6 on the security appliance. IPv6 is available in Routed firewall mode only.

This chapter includes the following sections:

- IPv6-enabled Commands, page 13-1
- Configuring IPv6, page 13-2
- Verifying the IPv6 Configuration, page 13-11

For an sample IPv6 configuration, see Appendix A, "Sample Configurations."

IPv6-enabled Commands

The following security appliance commands can accept and display IPv6 addresses:

- **capture**
- **configure**
- **copy**
- **http**
- **name**
- **object-group**
- **ping**
- **show conn**
- **show local-host**
- **show tcpstat**
- **ssh**
- **telnet**
- **tftp-server**
- **who**
- **write**

Note Failover does not support IPv6. The **ipv6 address** command does not support setting standby addresses for failover configurations. The **failover interface ip** command does not support using IPv6 addresses on the failover and Stateful Failover interfaces.

When entering IPv6 addresses in commands that support them, simply enter the IPv6 address using standard IPv6 notation, for example: `ping fe80::2e0:b6ff:fe01:3b7a`. The security appliance correctly recognizes and processes the IPv6 address. However, you must enclose the IPv6 address in square brackets ([ ]) in the following situations:

- You need to specify a port number with the address, for example: `[fe80::2e0:b6ff:fe01:3b7a]:8080`.
- The command uses a colon as a separator, such as the **write net** and **config net** commands, for example: `configure net [fe80::2e0:b6ff:fe01:3b7a]:/tftp/config/pixconfig`.

The following commands were modified to work for IPv6:

- **debug**
- **fragment**
- **ip verify**
- **mtu**
- **icmp** (entered as **ipv6 icmp**)

The following inspection engines support IPv6:

- FTP
- HTTP
- ICMP
- SIP
- SMTP
- TCP
- UDP

Configuring IPv6

This section contains the following topics:

- Configuring IPv6 on an Interface, page 13-3
- Configuring a Dual IP Stack on an Interface, page 13-4
- Enforcing the Use of Modified EUI-64 Interface IDs in IPv6 Addresses, page 13-4
- Configuring IPv6 Duplicate Address Detection, page 13-4
- Configuring IPv6 Default and Static Routes, page 13-5
- Configuring IPv6 Access Lists, page 13-6
- Configuring IPv6 Neighbor Discovery, page 13-7
- Configuring a Static IPv6 Neighbor, page 13-11

Configuring IPv6 on an Interface

At a minimum, each interface needs to be configured with an IPv6 link-local address. Additionally, you can add a global address to the interface.

Note The security appliance does not support IPv6 anycast addresses.

You can configure both IPv6 and IPv4 addresses on an interface.

To configure IPv6 on an interface, perform the following steps:

Step 1 Enter interface configuration mode for the interface on which you are configuring the IPv6 addresses:

```
hostname(config)# interface if
```

Step 2 Configure an IPv6 address on the interface. You can assign several IPv6 addresses to an interface, such as an IPv6 link-local and a global address. However, at a minimum, you must configure a link-local address.

There are several methods for configuring IPv6 addresses. Pick the method that suits your needs from the following:

- The simplest method is to enable stateless autoconfiguration on the interface. Enabling stateless autoconfiguration on the interface configures IPv6 addresses based on prefixes received in Router Advertisement messages. A link-local address, based on the Modified EUI-64 interface ID, is automatically generated for the interface when stateless autoconfiguration is enabled. To enable stateless autoconfiguration, enter the following command:

  ```
  hostname(config-if)# ipv6 address autoconfig
  ```

- If you only need to configure a link-local address on the interface and are not going to assign any other IPv6 addresses to the interface, you have the option of manually defining the link-local address or generating one based on the interface MAC address (Modified EUI-64 format):
  - Enter the following command to manually specify the link-local address:

    ```
    hostname(config-if)# ipv6 address ipv6-address link-local
    ```

  - Enter the following command to enable IPv6 on the interface and automatically generate the link-local address using the Modified EUI-64 interface ID based on the interface MAC address:

    ```
    hostname(config-if)# ipv6 enable
    ```

  Note You do not need to use the **ipv6 enable** command if you enter any other **ipv6 address** commands on an interface; IPv6 support is automatically enabled as soon as you assign an IPv6 address to the interface.

- Assign a global address to the interface. When you assign a global address, a link-local address is automatically created. Enter the following command to add a global to the interface. Use the optional **eui-64** keyword to use the Modified EUI-64 interface ID in the low order 64 bits of the address.

  ```
  hostname(config-if)# ipv6 address ipv6-prefix/prefix-length [eui-64]
  ```

Step 3 (Optional) Suppress Router Advertisement messages on an interface. By default, Router Advertisement messages are automatically sent in response to router solicitation messages. You may want to disable these messages on any interface for which you do not want the security appliance to supply the IPv6 prefix (for example, the outside interface).

Enter the following command to suppress Router Advertisement messages on an interface:

```
hostname(config-if)# ipv6 nd suppress-ra
```

Configuring a Dual IP Stack on an Interface

The security appliance supports the configuration of both IPv6 and IPv4 on an interface. You do not need to enter any special commands to do so; simply enter the IPv4 configuration commands and IPv6 configuration commands as you normally would. Make sure you configure a default route for both IPv4 and IPv6.

Enforcing the Use of Modified EUI-64 Interface IDs in IPv6 Addresses

RFC 3513: Internet Protocol Version 6 (IPv6) Addressing Architecture requires that the interface identifier portion of all unicast IPv6 addresses, except those that start with binary value 000, be 64 bits long and be constructed in Modified EUI-64 format. The security appliance can enforce this requirement for hosts attached to the local link.

To enforce the use of Modified EUI-64 format interface identifiers in IPv6 addresses on a local link, enter the following command:

```
hostname(config)# ipv6 enforce-eui64 if_name
```

The *if_name* argument is the name of the interface, as specified by the **nameif** command, on which you are enabling the address format enforcement.

When this command is enabled on an interface, the source addresses of IPv6 packets received on that interface are verified against the source MAC addresses to ensure that the interface identifiers use the Modified EUI-64 format. If the IPv6 packets do not use the Modified EUI-64 format for the interface identifier, the packets are dropped and the following system log message is generated:

```
%PIX|ASA-3-325003: EUI-64 source address check failed.
```

The address format verification is only performed when a flow is created. Packets from an existing flow are not checked. Additionally, the address verification can only be performed for hosts on the local link. Packets received from hosts behind a router will fail the address format verification, and be dropped, because their source MAC address will be the router MAC address and not the host MAC address.

Configuring IPv6 Duplicate Address Detection

During the stateless autoconfiguration process, duplicate address detection verifies the uniqueness of new unicast IPv6 addresses before the addresses are assigned to interfaces (the new addresses remain in a tentative state while duplicate address detection is performed). Duplicate address detection is performed first on the new link-local address. When the link local address is verified as unique, then duplicate address detection is performed all the other IPv6 unicast addresses on the interface.

Duplicate address detection is suspended on interfaces that are administratively down. While an interface is administratively down, the unicast IPv6 addresses assigned to the interface are set to a pending state. An interface returning to an administratively up state restarts duplicate address detection for all of the unicast IPv6 addresses on the interface.

When a duplicate address is identified, the state of the address is set to DUPLICATE, the address is not used, and the following error message is generated:

```
%PIX|ASA-4-325002: Duplicate address ipv6_address/MAC_address on interface
```

If the duplicate address is the link-local address of the interface, the processing of IPv6 packets is disabled on the interface. If the duplicate address is a global address, the address is not used. However, all configuration commands associated with the duplicate address remain as configured while the state of the address is set to DUPLICATE.

If the link-local address for an interface changes, duplicate address detection is performed on the new link-local address and all of the other IPv6 address associated with the interface are regenerated (duplicate address detection is performed only on the new link-local address).

The security appliance uses neighbor solicitation messages to perform duplicate address detection. By default, the number of times an interface performs duplicate address detection is 1.

To change the number of duplicate address detection attempts, enter the following command:

```
hostname(config-if)# ipv6 nd dad attempts value
```

The *value* argument can be any value from 0 to 600. Setting the *value* argument to 0 disables duplicate address detection on the interface.

When you configure an interface to send out more than one duplicate address detection attempt, you can also use the **ipv6 nd ns-interval** command to configure the interval at which the neighbor solicitation messages are sent out. By default, they are sent out once every 1000 milliseconds.

To change the neighbor solicitation message interval, enter the following command:

```
hostname(config-if)# ipv6 nd ns-interval value
```

The *value* argument can be from 1000 to 3600000 milliseconds.



Note Changing this value changes it for all neighbor solicitation messages sent out on the interface, not just those used for duplicate address detection.

Configuring IPv6 Default and Static Routes

The security appliance automatically routes IPv6 traffic between directly connected hosts if the interfaces to which the hosts are attached are enabled for IPv6 and the IPv6 ACLs allow the traffic.

The security appliance does not support dynamic routing protocols. Therefore, to route IPv6 traffic to a non-connected host or network, you need to define a static route to the host or network or, at a minimum, a default route. Without a static or default route defined, traffic to non-connected hosts or networks generate the following error message:

```
%PIX|ASA-6-110001: No route to dest_address from source_address
```

You can add a default route and static routes using the **ipv6 route** command.

To configure an IPv6 default route and static routes, perform the following steps:

Step 1 To add the default route, use the following command:

```
hostname(config)# ipv6 route if_name ::/0 next_hop_ipv6_addr
```

The address ::/0 is the IPv6 equivalent of "any."

Step 2 (Optional) Define IPv6 static routes. Use the following command to add an IPv6 static route to the IPv6 routing table:

```
hostname(config)# ipv6 route if_name destination next_hop_ipv6_addr [admin_distance]
```

Note The **ipv6 route** command works like the **route** command used to define IPv4 static routes.

Configuring IPv6 Access Lists

Configuring an IPv6 access list is similar configuring an IPv4 access, but with IPv6 addresses.

To configure an IPv6 access list, perform the following steps:

Step 1 Create an access entry. To create an access list, use the **ipv6 access-list** command to create entries for the access list. There are two main forms of this command to choose from, one for creating access list entries specifically for ICMP traffic, and one to create access list entries for all other types of IP traffic.

- To create an IPv6 access list entry specifically for ICMP traffic, enter the following command:

  ```
  hostname(config)# ipv6 access-list id [line num] {permit | deny} icmp source
  destination [icmp_type]
  ```

- To create an IPv6 access list entry, enter the following command:

  ```
  hostname(config)# ipv6 access-list id [line num] {permit | deny} protocol source
  [src_port] destination [dst_port]
  ```

The following describes the arguments for the **ipv6 access-list** command:

- *id—The name of the access list. Use the same id in each command when you are entering multiple entries for an access list.*
- **line** *num*—When adding an entry to an access list, you can specify the line number in the list where the entry should appear.
- **permit | deny**—Determines whether the specified traffic is blocked or allowed to pass.
- **icmp**—Indicates that the access list entry applies to ICMP traffic.
- *protocol*—Specifies the traffic being controlled by the access list entry. This can be the name (**ip**, **tcp**, or **udp**) or number (1-254) of an IP protocol. Alternatively, you can specify a protocol object group using **object-group** *grp_id*.
- *source and destination*—Specifies the source or destination of the traffic. The source or destination can be an IPv6 prefix, in the format *prefix/length*, to indicate a range of addresses, the keyword **any, to specify any address**, or a specific host designated by **host** *host_ipv6_addr*.

- *src_port and dst_port*—The source and destination port (or service) argument. Enter an operator (**lt** for less than, **gt** for greater than, **eq** for equal to, **neq** for not equal to, or **range** for an inclusive range) followed by a space and a port number (or two port numbers separated by a space for the **range** keyword).
- *icmp_type*—Specifies the ICMP message type being filtered by the access rule. The value can be a valid ICMP type number (from 0 to 155) or one of the ICMP type literals as shown in Appendix C, "Addresses, Protocols, and Ports". Alternatively, you can specify an ICMP object group using **object-group** *id*.

Step 2 To apply the access list to an interface, enter the following command:

```
hostname(config)# access-group access_list_name {in | out} interface if_name
```

Configuring IPv6 Neighbor Discovery

The IPv6 neighbor discovery process uses ICMPv6 messages and solicited-node multicast addresses to determine the link-layer address of a neighbor on the same network (local link), verify the reachability of a neighbor, and keep track of neighboring routers.

This section contains the following topics:

- Configuring Neighbor Solicitation Messages, page 13-7
- Configuring Router Advertisement Messages, page 13-9

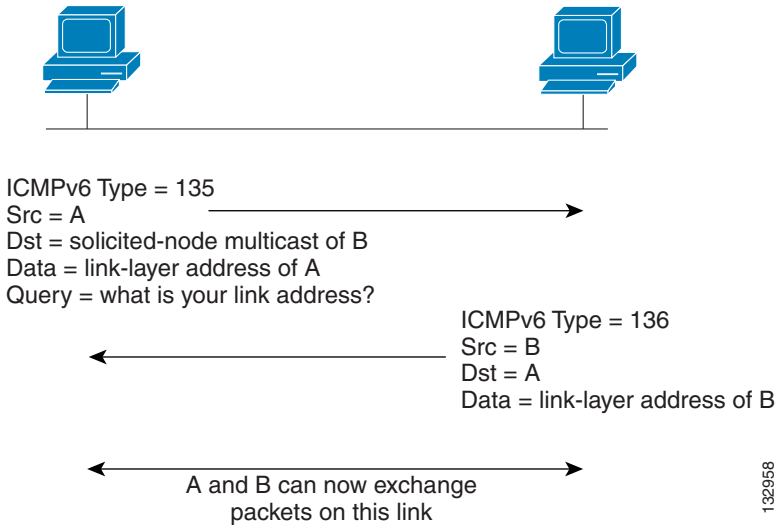
Configuring Neighbor Solicitation Messages

Neighbor solicitation messages (ICMPv6 Type 135) are sent on the local link by nodes attempting to discover the link-layer addresses of other nodes on the local link. The neighbor solicitation message is sent to the solicited-node multicast address.The source address in the neighbor solicitation message is the IPv6 address of the node sending the neighbor solicitation message. The neighbor solicitation message also includes the link-layer address of the source node.

After receiving a neighbor solicitation message, the destination node replies by sending a neighbor advertisement message (ICPMv6 Type 136) on the local link. The source address in the neighbor advertisement message is the IPv6 address of the node sending the neighbor advertisement message; the destination address is the IPv6 address of the node that sent the neighbor solicitation message. The data portion of the neighbor advertisement message includes the link-layer address of the node sending the neighbor advertisement message.

After the source node receives the neighbor advertisement, the source node and destination node can communicate. Figure 13-1 shows the neighbor solicitation and response process.

Figure 13-1 IPv6 Neighbor Discovery—Neighbor Solicitation Message



Neighbor solicitation messages are also used to verify the reachability of a neighbor after the link-layer address of a neighbor is identified. When a node wants to verifying the reachability of a neighbor, the destination address in a neighbor solicitation message is the unicast address of the neighbor.

Neighbor advertisement messages are also sent when there is a change in the link-layer address of a node on a local link. When there is such a change, the destination address for the neighbor advertisement is the all-nodes multicast address.

You can configure the neighbor solicitation message interval and neighbor reachable time on a per-interface basis. See the following topics for more information:

- Configuring the Neighbor Solicitation Message Interval, page 13-8
- Configuring the Neighbor Reachable Time, page 13-8

Configuring the Neighbor Solicitation Message Interval

To configure the interval between IPv6 neighbor solicitation retransmissions on an interface, enter the following command:

```
hostname(config-if)# ipv6 nd ns-interval value
```

Valid values for the *value* argument range from 1000 to 3600000 milliseconds. The default value is 1000 milliseconds.

This setting is also sent in router advertisement messages.

Configuring the Neighbor Reachable Time

The neighbor reachable time enables detecting unavailable neighbors. Shorter configured times enable detecting unavailable neighbors more quickly; however, shorter times consume more IPv6 network bandwidth and processing resources in all IPv6 network devices. Very short configured times are not recommended in normal IPv6 operation.

To configure the amount of time that a remote IPv6 node is considered reachable after a reachability confirmation event has occurred, enter the following command:

```
hostname(config-if)# ipv6 nd reachable-time value
```

Valid values for the *value* argument range from 0 to 3600000 milliseconds. The default is 0.

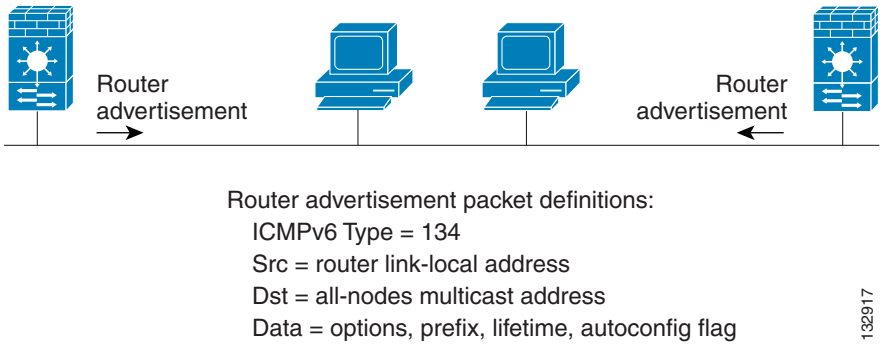
This information is also sent in router advertisement messages.

When 0 is used for the *value*, the reachable time is sent as undetermined. It is up to the receiving devices to set and track the reachable time value. To see the time used by the security appliance when this value is set to 0, use the **show ipv6 interface** command to display information about the IPv6 interface, including the ND reachable time being used.

Configuring Router Advertisement Messages

Router advertisement messages (ICMPv6 Type 134) are periodically sent out each IPv6 configured interface of the security appliance. The router advertisement messages are sent to the all-nodes multicast address.

Figure 13-2 IPv6 Neighbor Discovery—Router Advertisement Message



Router advertisement messages typically include the following information:

- One or more IPv6 prefix that nodes on the local link can use to automatically configure their IPv6 addresses.
- Lifetime information for each prefix included in the advertisement.
- Sets of flags that indicate the type of autoconfiguration (stateless or stateful) that can be completed.
- Default router information (whether the router sending the advertisement should be used as a default router and, if so, the amount of time (in seconds) the router should be used as a default router).
- Additional information for hosts, such as the hop limit and MTU a host should use in packets that it originates.
- The amount of time between neighbor solicitation message retransmissions on a given link.
- The amount of time a node considers a neighbor reachable.

Router advertisements are also sent in response to router solicitation messages (ICMPv6 Type 133). Router solicitation messages are sent by hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled router advertisement message. Because router solicitation messages are usually sent by hosts at system startup, and the host does not have a configured unicast address, the source address in router solicitation messages is usually the unspecified IPv6 address (0:0:0:0:0:0:0:0). If the host has a configured unicast address, the unicast address of the interface sending the router solicitation message is used as the source address in the message. The destination address in router solicitation messages is the all-routers multicast address with a scope of the link. When a router advertisement is sent in response to a router solicitation, the destination address in the router advertisement message is the unicast address of the source of the router solicitation message.

You can configure the following settings for router advertisement messages:

- The time interval between periodic router advertisement messages.
- The router lifetime value, which indicates the amount of time IPv6 nodes should consider the security appliance to be the default router.
- The IPv6 network prefixes in use on the link.
- Whether or not an interface transmits router advertisement messages.

Unless otherwise noted, the router advertisement message settings are specific to an interface and are entered in interface configuration mode. See the following topics for information about changing these settings:

- Configuring the Router Advertisement Transmission Interval, page 13-10
- Configuring the Router Lifetime Value, page 13-10
- Configuring the IPv6 Prefix, page 13-10
- Suppressing Router Advertisement Messages, page 13-11

Configuring the Router Advertisement Transmission Interval

By default, router advertisements are sent out every 200 seconds. To change the interval between router advertisement transmissions on an interface, enter the following command:

```
ipv6 nd ra-interval [msec] value
```

Valid values range from 3 to 1800 seconds (or 500 to 1800000 milliseconds if the **msec** keyword is used).

The interval between transmissions should be less than or equal to the IPv6 router advertisement lifetime if the security appliance is configured as a default router by using the **ipv6 nd ra-lifetime** command. To prevent synchronization with other IPv6 nodes, randomly adjust the actual value used to within 20 percent of the desired value.

Configuring the Router Lifetime Value

The router lifetime value specifies how long nodes on the local link should consider the security appliance the default router on the link.

To configure the router lifetime value in IPv6 router advertisements on an interface, enter the following command:

```
hostname(config-if)# ipv6 nd ra-lifetime seconds
```

Valid values range from 0 to 9000 seconds. The default is 1800 seconds. Entering 0 indicates that the security appliance should not be considered a default router on the selected interface.

Configuring the IPv6 Prefix

Stateless autoconfiguration uses IPv6 prefixes provided in router advertisement messages to create the global unicast address from the link-local address.

To configure which IPv6 prefixes are included in IPv6 router advertisements, enter the following command:

```
hostname(config-if)# ipv6 nd prefix ipv6-prefix/prefix-length
```

Note For stateless autoconfiguration to work properly, the advertised prefix length in router advertisement messages must always be 64 bits.

Suppressing Router Advertisement Messages

By default, Router Advertisement messages are automatically sent in response to router solicitation messages. You may want to disable these messages on any interface for which you do not want the security appliance to supply the IPv6 prefix (for example, the outside interface).

To suppress IPv6 router advertisement transmissions on an interface, enter the following command:

```
hostname(config-if)# ipv6 nd suppress-ra
```

Entering this command causes the security appliance to appear as a regular IPv6 neighbor on the link and not as an IPv6 router.

Configuring a Static IPv6 Neighbor

You can manually define a neighbor in the IPv6 neighbor cache. If an entry for the specified IPv6 address already exists in the neighbor discovery cache—learned through the IPv6 neighbor discovery process—the entry is automatically converted to a static entry. Static entries in the IPv6 neighbor discovery cache are not modified by the neighbor discovery process.

To configure a static entry in the IPv6 neighbor discovery cache, enter the following command:

```
hostname(config-if)# ipv6 neighbor ipv6_address if_name mac_address
```

The *ipv6_address* argument is the link-local IPv6 address of the neighbor, the *if_name* argument is the interface through which the neighbor is available, and the *mac_address* argument is the MAC address of the neighbor interface.

Note The **clear ipv6 neighbors** command does not remove static entries from the IPv6 neighbor discovery cache; it only clears the dynamic entries.

Verifying the IPv6 Configuration

This section describes how to verify your IPv6 configuration. You can use various show commands to verify your IPv6 settings.

This section includes the following topics:

- The show ipv6 interface Command, page 13-11
- The show ipv6 route Command, page 13-12

The show ipv6 interface Command

To display the IPv6 interface settings, enter the following command:

```
hostname# show ipv6 interface [if_name]
```

Including the interface name, such as “outside”, displays the settings for the specified interface. Excluding the name from the command displays the setting for all interfaces that have IPv6 enabled on them. The output for the command shows the following:

- The name and status of the interface.
- The link-local and global unicast addresses.

- The multicast groups the interface belongs to.
- ICMP redirect and error message settings.
- Neighbor discovery settings.

The following is sample output from the **show ipv6 interface** command:

```
hostname# show ipv6 interface

ipv6interface is down, line protocol is down
  IPv6 is enabled, link-local address is fe80::20d:88ff:feee:6a82 [TENTATIVE]
  No global unicast address is configured
  Joined group address(es):
    ff02::1
    ff02::1:ffee:6a82
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
```

Note The **show interface** command only displays the IPv4 settings for an interface. To see the IPv6 configuration on an interface, you need to use the **show ipv6 interface** command. The **show ipv6 interface** command does not display any IPv4 settings for the interface (if both types of addresses are configured on the interface).

The show ipv6 route Command

To display the routes in the IPv6 routing table, enter the following command:

```
hostname# show ipv6 route
```

The output from the **show ipv6 route** command is similar to the IPv4 **show route** command. It displays the following information:

- The protocol that derived the route.
- The IPv6 prefix of the remote network.
- The administrative distance and metric for the route.
- The address of the next-hop router.
- The interface through which the next hop router to the specified network is reached.

The following is sample output from the **show ipv6 route** command:

```
hostname# show ipv6 route

IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static
L   fe80::/10 [0/0]
     via ::, inside
L   fec0::a:0:0:a0a:a70/128 [0/0]
     via ::, inside
C   fec0:0:0:a::/64 [0/0]
     via ::, inside
L   ff00::/8 [0/0]
     via ::, inside
```



CHAPTER 14

Configuring AAA Servers and the Local Database

This chapter describes support for AAA (pronounced “triple A”) and how to configure AAA servers and the local database.

This chapter contains the following sections:

- AAA Overview, page 14-1
- AAA Server and Local Database Support, page 14-3
- Configuring the Local Database, page 14-7
- Identifying AAA Server Groups and Servers, page 14-9
- Configuring an LDAP Server, page 14-12
- Using Certificates and User Login Credentials, page 14-16
- Supporting a Zone Labs Integrity Server, page 14-17

AAA Overview

AAA enables the security appliance to determine who the user is (authentication), what the user can do (authorization), and what the user did (accounting).

AAA provides an extra level of protection and control for user access than using access lists alone. For example, you can create an access list allowing all outside users to access Telnet on a server on the DMZ network. If you want only some users to access the server and you might not always know IP addresses of these users, you can enable AAA to allow only authenticated and/or authorized users to make it through the security appliance. (The Telnet server enforces authentication, too; the security appliance prevents unauthorized users from attempting to access the server.)

You can use authentication alone or with authorization and accounting. Authorization always requires a user to be authenticated first. You can use accounting alone, or with authentication and authorization.

This section includes the following topics:

- About Authentication, page 14-2
- About Authorization, page 14-2
- About Accounting, page 14-2

About Authentication

Authentication controls access by requiring valid user credentials, which are typically a username and password. You can configure the security appliance to authenticate the following items:

- All administrative connections to the security appliance including the following sessions:
  - Telnet
  - SSH
  - Serial console
  - ASDM (using HTTPS)
  - VPN management access
- The **enable** command
- Network access
- VPN access

About Authorization

Authorization controls access *per user* after users authenticate. You can configure the security appliance to authorize the following items:

- Management commands
- Network access
- VPN access

Authorization controls the services and commands available to each authenticated user. Were you not to enable authorization, authentication alone would provide the same access to services for all authenticated users.

If you need the control that authorization provides, you can configure a broad authentication rule, and then have a detailed authorization configuration. For example, you authenticate inside users who attempt to access any server on the outside network and then limit the outside servers that a particular user can access using authorization.

The security appliance caches the first 16 authorization requests per user, so if the user accesses the same services during the current authentication session, the security appliance does not resend the request to the authorization server.

About Accounting

Accounting tracks traffic that passes through the security appliance, enabling you to have a record of user activity. If you enable authentication for that traffic, you can account for traffic per user. If you do not authenticate the traffic, you can account for traffic per IP address. Accounting information includes when sessions start and stop, username, the number of bytes that pass through the security appliance for the session, the service used, and the duration of each session.

AAA Server and Local Database Support

The security appliance supports a variety of AAA server types and a local database that is stored on the security appliance. This section describes support for each AAA server type and the local database.

This section contains the following topics:

- Summary of Support, page 14-3
- RADIUS Server Support, page 14-4
- TACACS+ Server Support, page 14-5
- RSA/SDI Server Support, page 14-5
- NT Server Support, page 14-6
- Kerberos Server Support, page 14-6
- LDAP Server Support, page 14-6
- SSO Support for Clientless SSL VPN with HTTP Forms, page 14-6
- Local Database Support, page 14-6

Summary of Support

Table 14-1 summarizes the support for each AAA service by each AAA server type, including the local database. For more information about support for a specific AAA server type, refer to the topics following the table.

Table 14-1 Summary of AAA Support

	Database Type							
AAA Service	**Local**	**RADIUS**	**TACACS+**	**SDI**	**NT**	**Kerberos**	**LDAP**	**HTTP Form**
Authentication of...								
VPN users[1]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes[2]
Firewall sessions	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Administrators	Yes	Yes	Yes	Yes[3]	Yes	Yes	Yes	No
Authorization of...								
VPN users	Yes	Yes	No	No	No	No	Yes	No
Firewall sessions	No	Yes[4]	Yes	No	No	No	No	No
Administrators	Yes[5]	No	Yes	No	No	No	No	No
Accounting of...								
VPN connections	No	Yes	Yes	No	No	No	No	No
Firewall sessions	No	Yes	Yes	No	No	No	No	No
Administrators	No	Yes[6]	Yes	No	No	No	No	No

1. For SSL VPN connections, either PAP or MS-CHAPv2 can be used.
2. HTTP Form protocol supports single sign-on authentication for Clientless SSL VPN users only.
3. SDI is not supported for HTTP administrative access.

4. For firewall sessions, RADIUS authorization is supported with user-specific access lists only, which are received or specified in a RADIUS authentication response.
5. Local command authorization is supported by privilege level only.
6. Command accounting is available for TACACS+ only.

RADIUS Server Support

The ASA supports the following RADIUS servers for AAA, in addition to the one available on the ASA itself:

- Cisco Secure ACS 3.2, 4.0, 4.1
- RSA Radius in RSA Authentication Manager 5.2 & 6.1

Authentication Methods

The security appliance supports the following authentication methods with RADIUS:

- PAP—For all connection types.
- CHAP—For L2TP-over-IPSec.
- MS-CHAPv1—For L2TP-over-IPSec.
- MS-CHAPv2—For L2TP-over-IPSec, and for regular IPSec remote access connections when the password-management feature is enabled. You can also use MS-CHAPv2 with Clientless connections.
- Authentication Proxy modes—Including RADIUS to Active Directory, RADIUS to RSA/SDI, RADIUS to Token-server, and RSA/SI to RADIUS,

Note To enable MSChapV2 as the protocol used between the security appliance and the RADIUS server for a clientless connection, password management must be enabled in the tunnel-group general-attributes. Enabling password management prevents usernames and passwords from being transmitted in clear text between the security appliance and the RADIUS server. See the description of the **password-management** command for details.

Attribute Support

The security appliance supports the following sets of RADIUS attributes:

- Authentication attributes defined in RFC 2138.
- Accounting attributes defined in RFC 2139.
- RADIUS attributes for tunneled protocol support, defined in RFC 2868.
- Cisco IOS VSAs, identified by RADIUS vendor ID 9.
- Cisco VPN-related VSAs, identified by RADIUS vendor ID 3076.
- Microsoft VSAs, defined in RFC 2548.

RADIUS Authorization Functions

The security appliance can use RADIUS servers for user authorization for network access using dynamic access lists or access list names per user. To implement dynamic access lists, you must configure the RADIUS server to support it. When the user authenticates, the RADIUS server sends a downloadable access list or access list name to the security appliance. Access to a given service is either permitted or denied by the access list. The security appliance deletes the access list when the authentication session expires.

TACACS+ Server Support

The security appliance supports TACACS+ authentication with ASCII, PAP, CHAP, and MS-CHAPv1.

RSA/SDI Server Support

The RSA SecureID servers are also known as SDI servers.

This section contains the following topics:

- RSA/SDI Version Support, page 14-5
- Two-step Authentication Process, page 14-5
- SDI Primary and Replica Servers, page 14-5

RSA/SDI Version Support

The security appliance supports SDI Version 5.0 and 6.0. SDI uses the concepts of an SDI primary and SDI replica servers. Each primary and its replicas share a single node secret file. The node secret file has its name based on the hexadecimal value of the ACE/Server IP address with .sdi appended.

A version 5.0 or 6.0 SDI server that you configure on the security appliance can be either the primary or any one of the replicas. See the "SDI Primary and Replica Servers" section on page 14-5 for information about how the SDI agent selects servers to authenticate users.

Two-step Authentication Process

SDI version 5.0 and 6.0 uses a two-step process to prevent an intruder from capturing information from an RSA SecurID authentication request and using it to authenticate to another server. The Agent first sends a lock request to the SecurID server before sending the user authentication request. The server locks the username, preventing another (replica) server from accepting it. This means that the same user cannot authenticate to two security appliances using the same authentication servers simultaneously. After a successful username lock, the security appliance sends the passcode.

SDI Primary and Replica Servers

The security appliance obtains the server list when the first user authenticates to the configured server, which can be either a primary or a replica. The security appliance then assigns priorities to each of the servers on the list, and subsequent server selection derives at random from those assigned priorities. The highest priority servers have a higher likelihood of being selected.

NT Server Support

The security appliance supports Microsoft Windows server operating systems that support NTLM version 1, collectively referred to as NT servers.

Note NT servers have a maximum length of 14 characters for user passwords. Longer passwords are truncated. This is a limitation of NTLM version 1.

Kerberos Server Support

The security appliance supports 3DES, DES, and RC4 encryption types.

Note The security appliance does not support changing user passwords during tunnel negotiation. To avoid this situation happening inadvertently, disable password expiration on the Kerberos/Active Directory server for users connecting to the security appliance.

For a simple Kerberos server configuration example, see Example 14-2 on page 14-12.

LDAP Server Support

The security appliance supports LDAP. For detailed information, see the "Configuring an LDAP Server" section on page 14-12.

SSO Support for Clientless SSL VPN with HTTP Forms

The security appliance can use the HTTP Form protocol for single sign-on (SSO) authentication of Clientless SSL VPN users only. Single sign-on support lets Clientless SSL VPN users enter a username and password only once to access multiple protected services and Web servers. The Clientless SSL VPN server running on the security appliance acts as a proxy for the user to the authenticating server. When a user logs in, the Clientless SSL VPN server sends an SSO authentication request, including username and password, to the authenticating server using HTTPS. If the server approves the authentication request, it returns an SSO authentication cookie to the Clientless SSL VPN server. The security appliance keeps this cookie on behalf of the user and uses it to authenticate the user to secure websites within the domain protected by the SSO server.

In addition to the HTTP Form protocol, Clientless SSL VPN administrators can choose to configure SSO with the HTTP Basic and NTLM authentication protocols (the **auto-signon** command), or with Computer Associates eTrust SiteMinder SSO server (formerly Netegrity SiteMinder) as well. For an in-depth discussion of configuring SSO with either HTTP Forms, **auto-signon** or SiteMinder, see the Configuring Clientless SSL VPN chapter.

Local Database Support

The security appliance maintains a local database that you can populate with user profiles.

This section contains the following topics:

- User Profiles, page 14-7
- Fallback Support, page 14-7

User Profiles

User profiles contain, at a minimum, a username. Typically, a password is assigned to each username, although passwords are optional.

The **username attributes** command lets you enter the username mode. In this mode, you can add other information to a specific user profile. The information you can add includes VPN-related attributes, such as a VPN session timeout value.

Fallback Support

The local database can act as a fallback method for several functions. This behavior is designed to help you prevent accidental lockout from the security appliance.

For users who need fallback support, we recommend that their usernames and passwords in the local database match their usernames and passwords in the AAA servers. This provides transparent fallback support. Because the user cannot determine whether a AAA server or the local database is providing the service, using usernames and passwords on AAA servers that are different than the usernames and passwords in the local database means that the user cannot be certain which username and password should be given.

The local database supports the following fallback functions:

- **Console and enable password authentication**—When you use the **aaa authentication console** command, you can add the **LOCAL** keyword after the AAA server group tag. If the servers in the group all are unavailable, the security appliance uses the local database to authenticate administrative access. This can include enable password authentication, too.
- **Command authorization**—When you use the **aaa authorization command** command, you can add the **LOCAL** keyword after the AAA server group tag. If the TACACS+ servers in the group all are unavailable, the local database is used to authorize commands based on privilege levels.
- **VPN authentication and authorization**—VPN authentication and authorization are supported to enable remote access to the security appliance if AAA servers that normally support these VPN services are unavailable. The **authentication-server-group** command, available in tunnel-group general attributes mode, lets you specify the **LOCAL** keyword when you are configuring attributes of a tunnel group. When VPN client of an administrator specifies a tunnel group configured to fallback to the local database, the VPN tunnel can be established even if the AAA server group is unavailable, provided that the local database is configured with the necessary attributes.

Configuring the Local Database

This section describes how to manage users in the local database. You can use the local database for CLI access authentication, privileged mode authentication, command authorization, network access authentication, and VPN authentication and authorization. You cannot use the local database for network access authorization. The local database does not support accounting.

For multiple context mode, you can configure usernames in the system execution space to provide individual logins using the **login** command; however, you cannot configure any **aaa** commands in the system execution space.

To define a user account in the local database, perform the following steps:

Step 1 To create the user account, enter the following command:

```
hostname(config)# username name {nopassword | password password [mschap]} [privilege priv_level]
```

where the *username* keyword is a string from 4 to 64 characters long.

The **password** *password* argument is a string from 3 to 16 characters long.

The **mschap** keyword specifies that the password is e converted to unicode and hashed using MD4 after you enter it. Use this keyword if users are authenticated using MSCHAPv1 or MSCHAPv2.

The **privilege** *level* argument sets the privilege level from 0 to 15. The default is 2. This privilege level is used with command authorization.

Caution If you do not use command authorization (the **aaa authorization command LOCAL** command), then the default level 2 allows management access to privileged EXEC mode. If you want to limit access to privileged EXEC mode, either set the privilege level to 0 or 1, or use the **service-type** command (see Step 4).

The **nopassword** keyword creates a user account with no password.

Note The **encrypted** and **nt-encrypted** keywords are typically for display only. When you define a password in the **username** command, the security appliance encrypts it when it saves it to the configuration for security purposes. When you enter the **show running-config** command, the **username** command does not show the actual password; it shows the encrypted password followed by the **encrypted** or **nt-encrypted** keyword (when you specify **mschap**). For example, if you enter the password "test," the **show running-config** display would appear to be something like the following:

```
username pat password DLaUiAX3l78qgoB5c7iVNw== nt-encrypted
```

The only time you would actually enter the **encrypted** or **nt-encrypted** keyword at the CLI is if you are cutting and pasting a configuration to another security appliance and you are using the same password.

Step 2 (Optional) To enforce user-specific access levels for users who authenticate for management access (see the **aaa authentication console LOCAL** command), enter the following command:

```
hostname(config)# aaa authorization exec authentication-server
```

This command enables management authorization for local users and for any users authenticated by RADIUS, LDAP, and TACACS+. See the "Limiting User CLI and ASDM Access with Management Authorization" section on page 42-7 for information about configuring a user on a AAA server to accommodate management authorization.

For a local user, configure the level of access using the **service-type** command as described in Step 4.

Step 3 (Optional) To configure username attributes, enter the following command:

```
hostname(config)# username username attributes
```

where the *username* argument is the username you created in Step 1.

Step 4 (Optional) If you configured management authorization in Step 2, enter the following command to configure the user level:

```
hostname(config-username)# service-type {admin | nas-prompt | remote-access}
```

where the **admin** keyword allows full access to any services specified by the **aaa authentication console LOCAL** commands. **admin** is the default.

The **nas-prompt** keyword allows access to the CLI when you configure the **aaa authentication {telnet | ssh | serial} console LOCAL** command, but denies ASDM configuration access if you configure the **aaa authentication http console LOCAL** command. ASDM monitoring access is allowed. If you configure enable authentication with the **aaa authentication enable console LOCAL** command, the user cannot access privileged EXEC mode using the **enable** command (or by using the **login** command).

The **remote-access** keyword denies management access. The user cannot use any services specified by the **aaa authentication console LOCAL** commands (excluding the **serial** keyword; serial access is allowed).

Step 5 (Optional) If you are using this username for VPN authentication, you can configure many VPN attributes for the user. See the "Configuring User Attributes" section on page 32-75.

For example, the following command assigns a privilege level of 15 to the admin user account:

```
hostname(config)# username admin password passw0rd privilege 15
```

The following command creates a user account with no password:

```
hostname(config)# username bcham34 nopassword
```

The following commands enable management authorization, creates a user account with a password, enters username attributes configuration mode, and specifies the service-type attribute:

```
hostname(config)# aaa authorization exec authentication-server
hostname(config)# username rwilliams password gOgeOus
hostname(config)# username rwilliams attributes
hostname(config-username)# service-type nas-prompt
```

Identifying AAA Server Groups and Servers

If you want to use an external AAA server for authentication, authorization, or accounting, you must first create at least one AAA server group per AAA protocol and add one or more servers to each group. You identify AAA server groups by name. Each server group is specific to one type of server: Kerberos, LDAP, NT, RADIUS, SDI, or TACACS+.

The security appliance contacts the first server in the group. If that server is unavailable, the security appliance contacts the next server in the group, if configured. If all servers in the group are unavailable, the security appliance tries the local database if you configured it as a fallback method (management authentication and authorization only). If you do not have a fallback method, the security appliance continues to try the AAA servers.

To create a server group and add AAA servers to it, follow these steps:

Step 1 For each AAA server group you need to create, follow these steps:

a. Identify the server group name and the protocol. To do so, enter the following command:

```
hostname(config)# aaa-server server_group protocol {kerberos | ldap | nt | radius |
sdi | tacacs+}
```

For example, to use RADIUS to authenticate network access and TACACS+ to authenticate CLI access, you need to create at least two server groups, one for RADIUS servers and one for TACACS+ servers.

You can have up to 15 single-mode server groups or 4 multi-mode server groups. Each server group can have up to 16 servers in single mode or up to 4 servers in multi-mode.

When you enter a **aaa-server protocol** command, you enter group mode.

b. If you want to specify the maximum number of requests sent to a AAA server in the group before trying the next server, enter the following command:

```
hostname(config-aaa-server-group)# max-failed-attempts number
```

The *number* can be between 1 and 5. The default is 3.

If you configured a fallback method using the local database (for management access only; see the "Configuring AAA for System Administrators" section on page 42-5 and the "Configuring TACACS+ Command Authorization" section on page 42-14 to configure the fallback mechanism), and all the servers in the group fail to respond, then the group is considered to be unresponsive, and the fallback method is tried. The server group remains marked as unresponsive for a period of 10 minutes (by default) so that additional AAA requests within that period do not attempt to contact the server group, and the fallback method is used immediately. To change the unresponsive period from the default, see the **reactivation-mode** command in the following step.

If you do not have a fallback method, the security appliance continues to retry the servers in the group.

c. If you want to specify the method (reactivation policy) by which failed servers in a group are reactivated, enter the following command:

```
hostname(config-aaa-server-group)# # reactivation-mode {depletion [deadtime minutes] |
timed}
```

Where the **depletion** keyword reactivates failed servers only after all of the servers in the group are inactive.

The **deadtime** *minutes* argument specifies the amount of time in minutes, between 0 and 1440, that elapses between the disabling of the last server in the group and the subsequent re-enabling of all servers. The default is 10 minutes.

The **timed** keyword reactivates failed servers after 30 seconds of down time.

d. If you want to send accounting messages to all servers in the group (RADIUS or TACACS+ only), enter the following command:

```
hostname(config-aaa-server-group)# accounting-mode simultaneous
```

To restore the default of sending messages only to the active server, enter the **accounting-mode single** command.

Step 2 For each AAA server on your network, follow these steps:

a. Identify the server, including the AAA server group it belongs to. To do so, enter the following command:

```
hostname(config)# aaa-server server_group (interface_name) host server_ip
```

When you enter a **aaa-server host** command, you enter host mode.

b. As needed, use host mode commands to further configure the AAA server.

The commands in host mode do not apply to all AAA server types. Table 14-2 lists the available commands, the server types they apply to, and whether a new AAA server definition has a default value for that command. Where a command is applicable to the server type you specified and no default value is provided (indicated by "—"), use the command to specify the value. For more information about these commands, see the *Cisco Security Appliance Command Reference*.

Table 14-2 Host Mode Commands, Server Types, and Defaults

Command	Applicable AAA Server Types	Default Value
accounting-port	RADIUS	1646
acl-netmask-convert	RADIUS	standard
authentication-port	RADIUS	1645
kerberos-realm	Kerberos	—
key	RADIUS	—
	TACACS+	—
ldap-attribute-map	LDAP	—
ldap-base-dn	LDAP	—
ldap-login-dn	LDAP	—
ldap-login-password	LDAP	—
ldap-naming-attribute	LDAP	—
ldap-over-ssl	LDAP	—
ldap-scope	LDAP	—
nt-auth-domain-controller	NT	—
radius-common-pw	RADIUS	—
retry-interval	Kerberos	10 seconds
	RADIUS	10 seconds
	SDI	10 seconds
sasl-mechanism	LDAP	—
server-port	Kerberos	88
	LDAP	389
	NT	139
	SDI	5500
	TACACS+	49
server-type	LDAP	auto-discovery
timeout	All	10 seconds

Example 14-1 shows commands that add one TACACS+ group with one primary and one backup server, one RADIUS group with a single server, and an NT domain server.

Example 14-1 Multiple AAA Server Groups and Servers

```
hostname(config)# aaa-server AuthInbound protocol tacacs+
hostname(config-aaa-server-group)# max-failed-attempts 2
hostname(config-aaa-server-group)# reactivation-mode depletion deadtime 20
hostname(config-aaa-server-group)# exit
hostname(config)# aaa-server AuthInbound (inside) host 10.1.1.1
hostname(config-aaa-server-host)# key TACPlusUauthKey
hostname(config-aaa-server-host)# exit
hostname(config)# aaa-server AuthInbound (inside) host 10.1.1.2
hostname(config-aaa-server-host)# key TACPlusUauthKey2
hostname(config-aaa-server-host)# exit
hostname(config)# aaa-server AuthOutbound protocol radius
hostname(config-aaa-server-group)# exit
hostname(config)# aaa-server AuthOutbound (inside) host 10.1.1.3
hostname(config-aaa-server-host)# key RadUauthKey
hostname(config-aaa-server-host)# exit
hostname(config)# aaa-server NTAuth protocol nt
hostname(config-aaa-server-group)# exit
hostname(config)# aaa-server NTAuth (inside) host 10.1.1.4
hostname(config-aaa-server-host)# nt-auth-domain-controller primary1
hostname(config-aaa-server-host)# exit
```

Example 14-2 shows commands that configure a Kerberos AAA server group named watchdogs, add a AAA server to the group, and define the Kerberos realm for the server. Because Example 14-2 does not define a retry interval or the port that the Kerberos server listens to, the security appliance uses the default values for these two server-specific parameters. Table 14-2 lists the default values for all AAA server host mode commands.

Note Kerberos realm names use numbers and upper-case letters only. Although the security appliance accepts lower-case letters for a realm name, it does not translate lower-case letters to upper-case letters. Be sure to use upper-case letters only.

Example 14-2 Kerberos Server Group and Server

```
hostname(config)# aaa-server watchdogs protocol kerberos
hostname(config-aaa-server-group)# aaa-server watchdogs host 192.168.3.4
hostname(config-aaa-server-host)# kerberos-realm EXAMPLE.COM
hostname(config-aaa-server-host)# exit
hostname(config)#
```

Configuring an LDAP Server

This section describes using an LDAP directory with the security appliance for user authentication and VPN authorization. This section includes the following topics:

- Authentication with LDAP, page 14-13
- Authorization with LDAP for VPN, page 14-14
- LDAP Attribute Mapping, page 14-15

For example configuration procedures used to set up LDAP authentication or authorization, see Appendix D, “Configuring an External Server for Authorization and Authentication”.

Authentication with LDAP

During authentication, the security appliance acts as a client proxy to the LDAP server for the user, and authenticates to the LDAP server in either plain text or using the Simple Authentication and Security Layer (SASL) protocol. By default, the security appliance passes authentication parameters, usually a username and password, to the LDAP server in plain text. Whether using SASL or plain text, you can secure the communications between the security appliance and the LDAP server with SSL using the **ldap-over-ssl** command.

Note If you do not configure SASL, we strongly recommend that you secure LDAP communications with SSL. See the **ldap-over-ssl** command in the *Cisco Security Appliance Command Reference*.

When user LDAP authentication has succeeded, the LDAP server returns the attributes for the authenticated user. For VPN authentication, these attributes generally include authorization data which is applied to the VPN session. Thus, using LDAP accomplishes authentication and authorization in a single step.

Securing LDAP Authentication with SASL

The security appliance supports the following SASL mechanisms, listed in order of increasing strength:

- Digest-MD5 — The security appliance responds to the LDAP server with an MD5 value computed from the username and password.
- Kerberos — The security appliance responds to the LDAP server by sending the username and realm using the GSSAPI (Generic Security Services Application Programming Interface) Kerberos mechanism.

You can configure the security appliance and LDAP server to support any combination of these SASL mechanisms. If you configure multiple mechanisms, the security appliance retrieves the list of SASL mechanisms configured on the server and sets the authentication mechanism to the strongest mechanism configured on both the security appliance and the server. For example, if both the LDAP server and the security appliance support both mechanisms, the security appliance selects Kerberos, the stronger of the mechanisms.

The following example configures the security appliance for authentication to an LDAP directory server named ldap_dir_1 using the digest-MD5 SASL mechanism, and communicating over an SSL-secured connection:

```
hostname(config)# aaa-server ldap_dir_1 protocol ldap
hostname(config-aaa-server-group)# aaa-server ldap_dir_1 host 10.1.1.4
hostname(config-aaa-server-host)# sasl-mechanism digest-md5
hostname(config-aaa-server-host)# ldap-over-ssl enable
hostname(config-aaa-server-host)#
```

Setting the LDAP Server Type

The security appliance supports LDAP version 3 and is compatible with the Sun Microsystems JAVA System Directory Server (formerly named the Sun ONE Directory Server), the Microsoft Active Directory, and other LDAPv3 directory servers.

By default, the security appliance auto-detects whether it is connected to a Microsoft Active Directory, a Sun LDAP directory server, or a generic LDAPv3 directory server. However, if auto-detection fails to determine the LDAP server type, and you know the server is either a Microsoft, Sun or generic LDAP server, you can manually configure the server type using the keywords **sun**, **microsoft**, or **generic**. The following example sets the LDAP directory server ldap_dir_1 to the Sun Microsystems type:

```
hostname(config)# aaa-server ldap_dir_1 protocol ldap
hostname(config-aaa-server-group)# aaa-server ldap_dir_1 host 10.1.1.4
hostname(config-aaa-server-host)# server-type sun
hostname(config-aaa-server-host)#
```

Note

- Sun—The DN configured on the security appliance to access a Sun directory server must be able to access the default password policy on that server. We recommend using the directory administrator, or a user with directory administrator privileges, as the DN. Alternatively, you can place an ACI on the default password policy.
- Microsoft—You must configure LDAP over SSL to enable password management with Microsoft Active Directory.
- Generic—The security appliance does not support password management with a generic LDAPv3 directory server.

Authorization with LDAP for VPN

When user LDAP authentication for VPN access has succeeded, the security appliance queries the LDAP server which returns LDAP attributes. These attributes generally include authorization data that applies to the VPN session. Thus, using LDAP accomplishes authentication and authorization in a single step.

There may be cases, however, where you require authorization from an LDAP directory server that is separate and distinct from the authentication mechanism. For example, if you use an SDI or certificate server for authentication, no authorization information is passed back. For user authorizations in this case, you can query an LDAP directory after successful authentication, accomplishing authentication and authorization in two steps.

To set up VPN user authorization using LDAP, you must first create a AAA server group and a tunnel group. You then associate the server and tunnel groups using the **tunnel-group general-attributes** command. While there are other authorization-related commands and options available for specific requirements, the following example shows fundamental commands for enabling user authorization with LDAP. This example then creates an IPSec remote access tunnel group named remote-1, and assigns that new tunnel group to the previously created ldap_dir_1 AAA server for authorization.

```
hostname(config)# tunnel-group remote-1 type ipsec-ra
hostname(config)# tunnel-group remote-1 general-attributes
hostname(config-general)# authorization-server-group ldap_dir_1
hostname(config-general)#
```

After you complete this fundamental configuration work, you can configure additional LDAP authorization parameters such as a directory password, a starting point for searching a directory, and the scope of a directory search:

```
hostname(config)# aaa-server ldap_dir_1 protocol ldap
hostname(config-aaa-server-group)# aaa-server ldap_dir_1 host 10.1.1.4
hostname(config-aaa-server-host)# ldap-login-dn obscurepassword
hostname(config-aaa-server-host)# ldap-base-dn starthere
hostname(config-aaa-server-host)# ldap-scope subtree
hostname(config-aaa-server-host)#
```

See LDAP commands in the *Cisco Security Appliance Command Reference* for more information.

LDAP Attribute Mapping

If you are introducing a security appliance to an existing LDAP directory, your existing LDAP attribute names and values are probably different from the existing ones. You must create LDAP attribute maps that map your existing user-defined attribute names and values to Cisco attribute names and values that are compatible with the security appliance. You can then bind these attribute maps to LDAP servers or remove them as needed. You can also show or clear attribute maps.

Note To use the attribute mapping features correctly, you need to understand the Cisco LDAP attribute names and values as well as the user-defined attribute names and values.

The following command, entered in global configuration mode, creates an unpopulated LDAP attribute map table named att_map_1:

```
hostname(config)# ldap attribute-map att_map_1
hostname(config-ldap-attribute-map)#
```

The following commands map the user-defined attribute name department to the Cisco attribute name IETF-Radius-Class. The second command maps the user-defined attribute value Engineering to the user-defined attribute department and the Cisco-defined attribute value group1.

```
hostname(config)# ldap attribute-map att_map_1
hostname(config-ldap-attribute-map)# map-name department IETF-Radius-Class
hostname(config-ldap-attribute-map)# map-value department Engineering group1
hostname(config-ldap-attribute-map)#
```

The following commands bind the attribute map att_map_1 to the LDAP server ldap_dir_1:

```
hostname(config)# aaa-server ldap_dir_1 host 10.1.1.4
hostname(config-aaa-server-host)# ldap-attribute-map att_map_1
hostname(config-aaa-server-host)#
```

Note The command to create an attribute map (**ldap attribute-map**) and the command to bind it to an LDAP server (**ldap-attribute-map**) differ only by a hyphen and the mode.

The following commands display or clear all LDAP attribute maps in the running configuration:

```
hostname# show running-config all ldap attribute-map
hostname(config)# clear configuration ldap attribute-map
hostname(config)#
```

The names of frequently mapped Cisco LDAP attributes and the type of user-defined attributes they would commonly be mapped to include:

IETF-Radius-Class — Department or user group
IETF-Radius-Filter-Id — Access control list
IETF-Radius-Framed-IP-Address — A static IP address
IPSec-Banner1 — A organization title
Tunneling-Protocols — Allow or deny dial-in

The following example shows how to limit management sessions to the security appliance based on an LDAP attribute called accessType. The accessType attribute has three possible values:

- VPN
- admin
- helpdesk

Each value is mapped to one of the valid IETF RADIUS Service-Types that the security appliance supports: remote-access (Service-Type 5) Outbound, admin (Service-Type 6) Administrative, and nas-prompt (Service-Type 7) NAS Prompt.

```
hostname(config)# ldap attribute-map MGMT
hostname(config-ldap-attribute-map)# map-name accessType IETF-Radius-Service-Type
hostname(config-ldap-attribute-map)# map-value accessType VPN 5
hostname(config-ldap-attribute-map)# map-value accessType admin 6
hostname(config-ldap-attribute-map)# map-value accessType helpdesk 7

hostname(config-ldap-attribute-map)# aaa-server LDAP protocol ldap
hostname(config-aaa-server-group)# aaa-server LDAP (inside) host 10.1.254.91
hostname(config-aaa-server-host)# ldap-base-dn CN=Users,DC=cisco,DC=local
hostname(config-aaa-server-host)# ldap-scope subtree
hostname(config-aaa-server-host)# ldap-login-password test
hostname(config-aaa-server-host)# ldap-login-dn
CN=Administrator,CN=Users,DC=cisco,DC=local
hostname(config-aaa-server-host)# server-type auto-detect
hostname(config-aaa-server-host)# ldap-attribute-map MGMT
```

For a list of Cisco LDAP attribute names and values, see Appendix D, "Configuring an External Server for Authorization and Authentication". Alternatively, you can enter "?" within ldap-attribute-map mode to display the complete list of Cisco LDAP attribute names, as shown in the following example:

```
hostname(config)# ldap attribute-map att_map_1
hostname(config-ldap-attribute-map)# map-name att_map_1 ?

ldap mode commands/options:
cisco-attribute-names:
Access-Hours
Allow-Network-Extension-Mode
Auth-Service-Type
Authenticated-User-Idle-Timeout
Authorization-Required
Authorization-Type
    :
    :
X509-Cert-Data
hostname(config-ldap-attribute-map)#
```

Using Certificates and User Login Credentials

The following section describes the different methods of using certificates and user login credentials (username and password) for authentication and authorization. This applies to both IPSec and Clientless SSL VPN.

In all cases, LDAP authorization does not use the password as a credential. RADIUS authorization uses either a common password for all users or the username as a password.

Using User Login Credentials

The default method for authentication and authorization uses the user login credentials.

- Authentication
  - Enabled by authentication server group setting
  - Uses the username and password as credentials

- Authorization
  - Enabled by authorization server group setting
  - Uses the username as a credential

Using certificates

If user digital certificates are configured, the security appliance first validates the certificate. It does not, however, use any of the DNs from the certificates as a username for the authentication.

If both authentication and authorization are enabled, the security appliance uses the user login credentials for both user authentication and authorization.

- Authentication
  - Enabled by authentication server group setting
  - Uses the username and password as credentials
- Authorization
  - Enabled by authorization server group setting
  - Uses the username as a credential

If authentication is disabled and authorization is enabled, the security appliance uses the primary DN field for authorization.

- Authentication
  - DISABLED (set to None) by authentication server group setting
  - No credentials used
- Authorization
  - Enabled by authorization server group setting
  - Uses the username value of the certificate primary DN field as a credential

Note If the primary DN field is not present in the certificate, the security appliance uses the secondary DN field value as the username for the authorization request.

For example, consider a user certificate that contains the following Subject DN fields and values:

`Cn=anyuser,OU=sales;O=XYZCorporation;L=boston;S=mass;C=us;ea=anyuser@example.com`.

If the Primary DN = EA (E-mail Address) and the Secondary DN = CN (Common Name), then the username used in the authorization request would be anyuser@example.com.

Supporting a Zone Labs Integrity Server

This section introduces the Zone Labs Integrity Server, also called Check Point Integrity Server, and presents an example procedure for configuring the security appliance to support the Zone Labs Integrity Server. The Integrity server is a central management station for configuring and enforcing security policies on remote PCs. If a remote PC does not conform to the security policy dictated by the Integrity Server, it will not be granted access to the private network protected by the Integrity Server and security appliance.

This section includes the following topics:

- Overview of Integrity Server and Security Appliance Interaction, page 14-18
- Configuring Integrity Server Support, page 14-18

Overview of Integrity Server and Security Appliance Interaction

The VPN client software and the Integrity client software are co-resident on a remote PC. The following steps summarize the actions of the remote PC, security appliance, and Integrity server in the establishment of a session between the PC and the enterprise private network:

1. The VPN client software (residing on the same remote PC as the Integrity client software) connects to the security appliance and tells the security appliance what type of firewall client it is.
2. Once it approves the client firewall type, the security appliance passes Integrity server address information back to the Integrity client.
3. With the security appliance acting as a proxy, the Integrity client establishes a restricted connection with the Integrity server. A restricted connection is only between the Integrity client and server.
4. The Integrity server determines if the Integrity client is in compliance with the mandated security policies. If the client is in compliance with security policies, the Integrity server instructs the security appliance to open the connection and provide the client with connection details.
5. On the remote PC, the VPN client passes connection details to the Integrity client and signals that policy enforcement should begin immediately and the client can no enter the private network.
6. Once the connection is established, the server continues to monitor the state of the client using client heartbeat messages.

Note The current release of the security appliance supports one Integrity Server at a time even though the user interfaces support the configuration of up to five Integrity Servers. If the active Server fails, configure another Integrity Server on the security appliance and then reestablish the client VPN session.

Configuring Integrity Server Support

This section describes an example procedure for configuring the security appliance to support the Zone Labs Integrity Servers. The procedure involves configuring address, port, connection fail timeout and fail states, and SSL certificate parameters.

First, you must configure the hostname or IP address of the Integrity server. The following example commands, entered in global configuration mode, configure an Integrity server using the IP address 10.0.0.5. They also specify port 300 (the default port is 5054) and the inside interface for communications with the Integrity server.

```
hostname(config)# zonelabs-integrity server-address 10.0.0.5
hostname(config)# zonelabs-integrity port 300
hostname(config)# zonelabs-integrity interface inside
hostname(config)#
```

If the connection between the security appliance and the Integrity server fails, the VPN client connections remain open by default so that the enterprise VPN is not disrupted by the failure of an Integrity server. However, you may want to close the VPN connections if the Zone Labs Integrity Server

fails. The following commands ensure that the security appliance waits 12 seconds for a response from either the active or standby Integrity servers before declaring an the Integrity server as failed and closing the VPN client connections:

```
hostname(config)# zonelabs-integrity fail-timeout 12
hostname(config)# zonelabs-integrity fail-close
hostname(config)#
```

The following command returns the configured VPN client connection fail state to the default and ensures the client connections remain open:

```
hostname(config)# zonelabs-integrity fail-open
hostname(config)#
```

The following example commands specify that the Integrity server connects to port 300 (default is port 80) on the security appliance to request the server SSL certificate. While the server SSL certificate is always authenticated, these commands also specify that the client SSL certificate of the Integrity server be authenticated.

```
hostname(config)# zonelabs-integrity ssl-certificate-port 300
hostname(config)# zonelabs-integrity ssl-client-authentication
hostname(config)#
```

To set the firewall client type to the Zone Labs Integrity type, use the **client-firewall** command as described in the "Configuring Firewall Policies" section on page 32-60. The command arguments that specify firewall policies are not used when the firewall type is **zonelabs-integrity** because the Integrity server determines the policies.



CHAPTER 15

Configuring Failover

This chapter describes the security appliance failover feature, which lets you configure two security appliances so that one takes over operation if the other one fails.

This chapter includes the following sections:

- Understanding Failover, page 15-1
- Configuring Failover, page 15-20
- Controlling and Monitoring Failover, page 15-51
- Remote Command Execution, page 15-53
- Auto Update Server Support in Failover Configurations, page 15-56

For failover configuration examples, see Appendix A, "Sample Configurations."

Understanding Failover

The failover configuration requires two identical security appliances connected to each other through a dedicated and, optionally, a Stateful Failover link. The health of the active interfaces and units is monitored to determine if specific failover conditions are met. If those conditions are met, failover occurs.

The security appliance supports two failover configurations, Active/Active failover and Active/Standby failover. Each failover configuration has its own method for determining and performing failover.

With Active/Active failover, both units can pass network traffic. This also lets you configure traffic sharing on your network. Active/Active failover is available only on units running in multiple context mode.

With Active/Standby failover, only one unit passes traffic while the other unit waits in a standby state. Active/Standby failover is available on units running in either single or multiple context mode.

Both failover configurations support stateful or stateless (regular) failover.

Note When the security appliance is configured for Active/Active stateful failover, you cannot enable IPSec or SSL VPN. Therefore, these features are unavailable. VPN failover is available for Active/Standby failover configurations only.

This section includes the following topics:

Failover System Requirements, page 15-2

- The Failover and Stateful Failover Links, page 15-3
- Active/Active and Active/Standby Failover, page 15-6
- Stateless (Regular) and Stateful Failover, page 15-16
- Failover Health Monitoring, page 15-18
- Failover Feature/Platform Matrix, page 15-19
- Failover Times by Platform, page 15-20

Failover System Requirements

This section describes the hardware, software, and license requirements for security appliances in a failover configuration. This section contains the following topics:

- Hardware Requirements, page 15-2
- Software Requirements, page 15-2
- License Requirements, page 15-3

Hardware Requirements

The two units in a failover configuration must have the same hardware configuration. They must be the same model, have the same number and types of interfaces, the same amount of RAM, and, for the ASA 5500 series security appliance, the same SSMs installed (if any).

Note The two units do not have to have the same size Flash memory. If using units with different Flash memory sizes in your failover configuration, make sure the unit with the smaller Flash memory has enough space to accommodate the software image files and the configuration files. If it does not, configuration synchronization from the unit with the larger Flash memory to the unit with the smaller Flash memory will fail.

Software Requirements

The two units in a failover configuration must be in the operating modes (routed or transparent, single or multiple context). They have the same major (first number) and minor (second number) software version. However, you can use different versions of the software during an upgrade process; for example, you can upgrade one unit from Version 7.0(1) to Version 7.0(2) and have failover remain active. We recommend upgrading both units to the same version to ensure long-term compatibility.

See "Performing Zero Downtime Upgrades for Failover Pairs" section on page 43-5 for more information about upgrading the software on a failover pair.

License Requirements

On the PIX 500 series security appliance, at least one of the units must have an unrestricted (UR) license. The other unit can have a Failover Only (FO) license, a Failover Only Active-Active (FO_AA) license, or another UR license. Units with a Restricted license cannot be used for failover, and two units with FO or FO_AA licenses cannot be used together as a failover pair.

Note The FO license does not support Active/Active failover.

The FO and FO_AA licenses are intended to be used solely for units in a failover configuration and not for units in standalone mode. If a failover unit with one of these licenses is used in standalone mode, the unit reboots at least once every 24 hours until the unit is returned to failover duty. A unit with an FO or FO_AA license operates in standalone mode if it is booted without being connected to a failover peer with a UR license. If the unit with a UR license in a failover pair fails and is removed from the configuration, the unit with the FO or FO_AA license does not automatically reboot every 24 hours; it operates uninterrupted unless the it is manually rebooted.

When the unit automatically reboots, the following message displays on the console:

```
=========================NOTICE=========================
       This machine is running in secondary mode without
         a connection to an active primary PIX. Please
           check your connection to the primary system.

                          REBOOTING....
========================================================
```

The ASA 5500 series adaptive security appliance platform does not have this restriction.

Note The licensed features (such as SSL VPN peers or security contexts, for example) on both security appliances participating in failover must be identical.

The Failover and Stateful Failover Links

This section describes the failover and the Stateful Failover links, which are dedicated connections between the two units in a failover configuration. This section includes the following topics:

- Failover Link, page 15-3
- Stateful Failover Link, page 15-5

Failover Link

The two units in a failover pair constantly communicate over a failover link to determine the operating status of each unit. The following information is communicated over the failover link:

- The unit state (active or standby).
- Power status (cable-based failover only—available only on the PIX 500 series security appliance).
- Hello messages (keep-alives).
- Network link status.

- MAC address exchange.
- Configuration replication and synchronization.

Caution All information sent over the failover and Stateful Failover links is sent in clear text unless you secure the communication with a failover key. If the security appliance is used to terminate VPN tunnels, this information includes any usernames, passwords and preshared keys used for establishing the tunnels. Transmitting this sensitive data in clear text could pose a significant security risk. We recommend securing the failover communication with a failover key if you are using the security appliance to terminate VPN tunnels.

On the PIX 500 series security appliance, the failover link can be either a LAN-based connection or a dedicated serial Failover cable. On the ASA 5500 series adaptive security appliance, the failover link can only be a LAN-based connection.

This section includes the following topics:

- LAN-Based Failover Link, page 15-4
- Serial Cable Failover Link (PIX Security Appliance Only), page 15-4

LAN-Based Failover Link

You can use any unused Ethernet interface on the device as the failover link; however, you cannot specify an interface that is currently configured with a name. The LAN failover link interface is not configured as a normal networking interface. It exists for failover communication only. This interface should only be used for the LAN failover link (and optionally for the stateful failover link).

Connect the LAN failover link in one of the following two ways:

- Using a switch, with no other device on the same network segment (broadcast domain or VLAN) as the LAN failover interfaces of the ASA.
- Using a crossover Ethernet cable to connect the appliances directly, without the need for an external switch.

Note When you use a crossover cable for the LAN failover link, if the LAN interface fails, the link is brought down on both peers. This condition may hamper troubleshooting efforts because you cannot easily determine which interface failed and caused the link to come down.

Note The ASA supports Auto-MDI/MDIX on its copper Ethernet ports, so you can either use a crossover cable or a straight-through cable. If you use a straight-through cable, the interface automatically detects the cable and swaps one of the transmit/receive pairs to MDIX.

Serial Cable Failover Link (PIX Security Appliance Only)

The serial Failover cable, or “cable-based failover,” is only available on the PIX 500 series security appliance. If the two units are within six feet of each other, then we recommend that you use the serial Failover cable.

The cable that connects the two units is a modified RS-232 serial link cable that transfers data at 117,760 bps (115 Kbps). One end of the cable is labeled “Primary”. The unit attached to this end of the cable automatically becomes the primary unit. The other end of the cable is labeled “Secondary”. The

unit attached to this end of the cable automatically becomes the secondary unit. You cannot override these designations in the PIX 500 series security appliance software. If you purchased a PIX 500 series security appliance failover bundle, this cable is included. To order a spare, use part number PIX-FO=.

The benefits of using cable-based failover include:

- The PIX 500 series security appliance can immediately detect a power loss on the peer unit and differentiate between a power loss from an unplugged cable.
- The standby unit can communicate with the active unit and can receive the entire configuration without having to be bootstrapped for failover. In LAN-based failover you need to configure the failover link on the standby unit before it can communicate with the active unit.
- The switch between the two units in LAN-based failover can be another point of hardware failure; cable-based failover eliminates this potential point of failure.
- You do not have to dedicate an Ethernet interface (and switch) to the failover link.
- The cable determines which unit is primary and which is secondary, eliminating the need to manually enter that information in the unit configurations.

The disadvantages include:

- Distance limitation—the units cannot be separated by more than 6 feet.
- Slower configuration replication.

Stateful Failover Link

To use Stateful Failover, you must configure a Stateful Failover link to pass all state information. You have three options for configuring a Stateful Failover link:

- You can use a dedicated Ethernet interface for the Stateful Failover link.
- If you are using LAN-based failover, you can share the failover link.
- You can share a regular data interface, such as the inside interface. However, this option is not recommended.

If you are using a dedicated Ethernet interface for the Stateful Failover link, you can use either a switch or a crossover cable to directly connect the units. If you use a switch, no other hosts or routers should be on this link.

Note Enable the PortFast option on Cisco switch ports that connect directly to the security appliance.

If you use a data interface as the Stateful Failover link, you receive the following warning when you specify that interface as the Stateful Failover link:

```
******* WARNING ***** WARNING ******* WARNING ****** WARNING  *********
  Sharing Stateful failover interface with regular data interface is not
  a recommended configuration due to performance and security concerns.
******* WARNING ***** WARNING ******* WARNING ****** WARNING  *********
```

Sharing a data interface with the Stateful Failover interface can leave you vulnerable to replay attacks. Additionally, large amounts of Stateful Failover traffic may be sent on the interface, causing performance problems on that network segment.

Note Using a data interface as the Stateful Failover interface is only supported in single context, routed mode.

In multiple context mode, the Stateful Failover link resides in the system context. This interface and the failover interface are the only interfaces in the system context. All other interfaces are allocated to and configured from within security contexts.

Note The IP address and MAC address for the Stateful Failover link does not change at failover unless the Stateful Failover link is configured on a regular data interface.

Caution All information sent over the failover and Stateful Failover links is sent in clear text unless you secure the communication with a failover key. If the security appliance is used to terminate VPN tunnels, this information includes any usernames, passwords and preshared keys used for establishing the tunnels. Transmitting this sensitive data in clear text could pose a significant security risk. We recommend securing the failover communication with a failover key if you are using the security appliance to terminate VPN tunnels.

Failover Interface Speed for Stateful Links

If you use the failover link as the Stateful Failover link, you should use the fastest Ethernet interface available. If you experience performance problems on that interface, consider dedicating a separate interface for the Stateful Failover interface.

Use the following failover interface speed guidelines for Cisco PIX security appliances and Cisco ASA adaptive security appliances:

- Cisco ASA 5520/5540/5550 and PIX 515E/535
  - The stateful link speed should match the fastest data link
- Cisco ASA 5510 and PIX 525
  - Stateful link speed can be 100 Mbps, even though the data interface can operate at 1 Gigabit due to the CPU speed limitation.

For optimum performance when using long distance LAN failover, the latency for the failover link should be less than 10 milliseconds and no more than 250 milliseconds. If latency is more than 10 milliseconds, some performance degradation occurs due to retransmission of failover messages.

All platforms support sharing of failover heartbeat and stateful link, but we recommend using a separate heartbeat link on systems with high Stateful Failover traffic.

Active/Active and Active/Standby Failover

This section describes each failover configuration in detail. This section includes the following topics:

- Active/Standby Failover, page 15-7
- Active/Active Failover, page 15-11
- Determining Which Type of Failover to Use, page 15-15

Active/Standby Failover

This section describes Active/Standby failover and includes the following topics:

- Active/Standby Failover Overview, page 15-7
- Primary/Secondary Status and Active/Standby Status, page 15-7
- Device Initialization and Configuration Synchronization, page 15-8
- Command Replication, page 15-8
- Failover Triggers, page 15-10
- Failover Actions, page 15-10

Active/Standby Failover Overview

Active/Standby failover lets you use a standby security appliance to take over the functionality of a failed unit. When the active unit fails, it changes to the standby state while the standby unit changes to the active state. The unit that becomes active assumes the IP addresses (or, for transparent firewall, the management IP address) and MAC addresses of the failed unit and begins passing traffic. The unit that is now in standby state takes over the standby IP addresses and MAC addresses. Because network devices see no change in the MAC to IP address pairing, no ARP entries change or time out anywhere on the network.

Note During a successful failover event on the security appliance, the interfaces are brought down, roles are switched (IP addresses and MAC addresses are swapped), and the interfaces are brought up again. However, the process is transparent to users. The security appliance does not send link-down messages or system log messages to notify users that interfaces were taken down during failover (or link-up messages for interfaces brought up by the failover process).

Note For multiple context mode, the security appliance can fail over the entire unit (including all contexts) but cannot fail over individual contexts separately.

Primary/Secondary Status and Active/Standby Status

The main differences between the two units in a failover pair are related to which unit is active and which unit is standby, namely which IP addresses to use and which unit actively passes traffic.

However, a few differences exist between the units based on which unit is primary (as specified in the configuration) and which unit is secondary:

- The primary unit always becomes the active unit if both units start up at the same time (and are of equal operational health).
- The primary unit MAC addresses are always coupled with the active IP addresses. The exception to this rule occurs when the secondary unit is active, and cannot obtain the primary unit MAC addresses over the failover link. In this case, the secondary unit MAC addresses are used.

Device Initialization and Configuration Synchronization

Configuration synchronization occurs when one or both devices in the failover pair boot. Configurations are always synchronized from the active unit to the standby unit. When the standby unit completes its initial startup, it clears its running configuration (except for the failover commands needed to communicate with the active unit), and the active unit sends its entire configuration to the standby unit.

The active unit is determined by the following:

- If a unit boots and detects a peer already running as active, it becomes the standby unit.
- If a unit boots and does not detect a peer, it becomes the active unit.
- If both units boot simultaneously, then the primary unit becomes the active unit and the secondary unit becomes the standby unit.

Note If the secondary unit boots without detecting the primary unit, it becomes the active unit. It uses its own MAC addresses for the active IP addresses. However, when the primary unit becomes available, the secondary unit changes the MAC addresses to those of the primary unit, which can cause an interruption in your network traffic. To avoid this, configure the failover pair with virtual MAC addresses. See the "Configuring Virtual MAC Addresses" section on page 15-28 for more information.

When the replication starts, the security appliance console on the active unit displays the message "Beginning configuration replication: Sending to mate," and when it is complete, the security appliance displays the message "End Configuration Replication to mate." During replication, commands entered on the active unit may not replicate properly to the standby unit, and commands entered on the standby unit may be overwritten by the configuration being replicated from the active unit. Avoid entering commands on either unit in the failover pair during the configuration replication process. Depending upon the size of the configuration, replication can take from a few seconds to several minutes.

Note The **crypto ca server** command and related sub-commands are not synchronized to the failover peer.

On the standby unit, the configuration exists only in running memory. To save the configuration to Flash memory after synchronization:

- For single context mode, enter the **write memory** command on the active unit. The command is replicated to the standby unit, which proceeds to write its configuration to Flash memory.
- For multiple context mode, enter the **write memory all** command on the active unit from the system execution space. The command is replicated to the standby unit, which proceeds to write its configuration to Flash memory. Using the **all** keyword with this command causes the system and all context configurations to be saved.

Note Startup configurations saved on external servers are accessible from either unit over the network and do not need to be saved separately for each unit. Alternatively, you can copy the contexts on disk from the active unit to an external server, and then copy them to disk on the standby unit, where they become available when the unit reloads.

Command Replication

Command replication always flows from the active unit to the standby unit. As commands are entered on the active unit, they are sent across the failover link to the standby unit. You do not have to save the active configuration to Flash memory to replicate the commands.

Table 15-1 lists the commands that are and are not replicated to the standby unit.

Table 15-1 Command Replication

Commands Replicated to the Standby Unit	Commands Not Replicated to the Standby Unit
all configuration commands except for the **mode**, **firewall**, and **failover lan unit** commands	all forms of the **copy** command except for **copy running-config startup-config**
copy running-config startup-config	all forms of the **write** command except for **write memory**
delete	**crypto ca server** and associated sub-commands
mkdir	**debug**
rename	**failover lan unit**
rmdir	**firewall**
write memory	**mode**
—	**show**
—	**terminal pager** and **pager**

Note Changes made on the standby unit are not replicated to the active unit. If you enter a command on the standby unit, the security appliance displays the message `**** WARNING **** Configuration Replication is NOT performed from Standby unit to Active unit. Configurations are no longer synchronized.` This message displays even when you enter many commands that do not affect the configuration.

If you enter the **write standby** command on the active unit, the standby unit clears its running configuration (except for the failover commands used to communicate with the active unit), and the active unit sends its entire configuration to the standby unit.

For multiple context mode, when you enter the **write standby** command in the system execution space, all contexts are replicated. If you enter the **write standby** command within a context, the command replicates only the context configuration.

Replicated commands are stored in the running configuration. To save the replicated commands to the Flash memory on the standby unit:

- For single context mode, enter the **copy running-config startup-config** command on the active unit. The command is replicated to the standby unit, which proceeds to write its configuration to Flash memory.
- For multiple context mode, enter the **copy running-config startup-config** command on the active unit from the system execution space and within each context on disk. The command is replicated to the standby unit, which proceeds to write its configuration to Flash memory. Contexts with startup configurations on external servers are accessible from either unit over the network and do not need to be saved separately for each unit. Alternatively, you can copy the contexts on disk from the active unit to an external server, and then copy them to disk on the standby unit.

Failover Triggers

The unit can fail if one of the following events occurs:

- The unit has a hardware failure or a power failure.
- The unit has a software failure.
- Too many monitored interfaces fail.
- The **no failover active** command is entered on the active unit or the **failover active** command is entered on the standby unit.

Failover Actions

In Active/Standby failover, failover occurs on a unit basis. Even on systems running in multiple context mode, you cannot fail over individual or groups of contexts.

Table 15-2 shows the failover action for each failure event. For each failure event, the table shows the failover policy (failover or no failover), the action taken by the active unit, the action taken by the standby unit, and any special notes about the failover condition and actions.

Table 15-2 Failover Behavior

Failure Event	Policy	Active Action	Standby Action	Notes
Active unit failed (power or hardware)	Failover	n/a	Become active Mark active as failed	No hello messages are received on any monitored interface or the failover link.
Formerly active unit recovers	No failover	Become standby	No action	None.
Standby unit failed (power or hardware)	No failover	Mark standby as failed	n/a	When the standby unit is marked as failed, then the active unit does not attempt to fail over, even if the interface failure threshold is surpassed.
Failover link failed during operation	No failover	Mark failover interface as failed	Mark failover interface as failed	You should restore the failover link as soon as possible because the unit cannot fail over to the standby unit while the failover link is down.
Failover link failed at startup	No failover	Mark failover interface as failed	Become active	If the failover link is down at startup, both units become active.
Stateful Failover link failed	No failover	No action	No action	State information becomes out of date, and sessions are terminated if a failover occurs.
Interface failure on active unit above threshold	Failover	Mark active as failed	Become active	None.
Interface failure on standby unit above threshold	No failover	No action	Mark standby as failed	When the standby unit is marked as failed, then the active unit does not attempt to fail over even if the interface failure threshold is surpassed.

Active/Active Failover

This section describes Active/Active failover. This section includes the following topics:

- Active/Active Failover Overview, page 15-11
- Primary/Secondary Status and Active/Standby Status, page 15-12
- Device Initialization and Configuration Synchronization, page 15-12
- Command Replication, page 15-13
- Failover Triggers, page 15-14
- Failover Actions, page 15-14

Active/Active Failover Overview

Active/Active failover is only available to security appliances in multiple context mode. In an Active/Active failover configuration, both security appliances can pass network traffic.

In Active/Active failover, you divide the security contexts on the security appliance into *failover groups*. A failover group is simply a logical group of one or more security contexts. You can create a maximum of two failover groups on the security appliance. The admin context is always a member of failover group 1. Any unassigned security contexts are also members of failover group 1 by default.

The failover group forms the base unit for failover in Active/Active failover. Interface failure monitoring, failover, and active/standby status are all attributes of a failover group rather than the unit. When an active failover group fails, it changes to the standby state while the standby failover group becomes active. The interfaces in the failover group that becomes active assume the MAC and IP addresses of the interfaces in the failover group that failed. The interfaces in the failover group that is now in the standby state take over the standby MAC and IP addresses.

Note During a successful failover event on the security appliance, the interfaces are brought down, roles are switched (IP addresses and MAC addresses are swapped), and the interfaces are brought up again. However, the security appliance does not send link-down messages or system log messages to notify users that interfaces were taken down during failover (or link-up messages for interfaces brought up by the failover process).

Note A failover group failing on a unit does not mean that the unit has failed. The unit may still have another failover group passing traffic on it.

When creating the failover groups, you should create them on the unit that will have failover group 1 in the active state.

Note Active/Active failover generates virtual MAC addresses for the interfaces in each failover group. If you have more than one Active/Active failover pair on the same network, it is possible to have the same default virtual MAC addresses assigned to the interfaces on one pair as are assigned to the interfaces of the other pairs because of the way the default virtual MAC addresses are determined. To avoid having duplicate MAC addresses on your network, make sure you assign each physical interface a virtual active and standby MAC address.

Primary/Secondary Status and Active/Standby Status

As in Active/Standby failover, one unit in an Active/Active failover pair is designated the primary unit, and the other unit the secondary unit. Unlike Active/Standby failover, this designation does not indicate which unit becomes active when both units start simultaneously. Instead, the primary/secondary designation does two things:

- Determines which unit provides the running configuration to the pair when they boot simultaneously.
- Determines on which unit each failover group appears in the active state when the units boot simultaneously. Each failover group in the configuration is configured with a primary or secondary unit preference. You can configure both failover groups be in the active state on a single unit in the pair, with the other unit containing the failover groups in the standby state. However, a more typical configuration is to assign each failover group a different role preference to make each one active on a different unit, distributing the traffic across the devices.

Note The security appliance also provides load balancing, which is different from failover. Both failover and load balancing can exist on the same configuration. For information about load balancing, see Understanding Load Balancing, page 31-6.

Which unit each failover group becomes active on is determined as follows:

- When a unit boots while the peer unit is not available, both failover groups become active on the unit.
- When a unit boots while the peer unit is active (with both failover groups in the active state), the failover groups remain in the active state on the active unit regardless of the primary or secondary preference of the failover group until one of the following:
  - A failover occurs.
  - You manually force the failover group to the other unit with the **no failover active** command.
  - You configured the failover group with the **preempt** command, which causes the failover group to automatically become active on the preferred unit when the unit becomes available.
- When both units boot at the same time, each failover group becomes active on its preferred unit after the configurations have been synchronized.

Device Initialization and Configuration Synchronization

Configuration synchronization occurs when one or both units in a failover pair boot. The configurations are synchronized as follows:

- When a unit boots while the peer unit is active (with both failover groups active on it), the booting unit contacts the active unit to obtain the running configuration regardless of the primary or secondary designation of the booting unit.
- When both units boot simultaneously, the secondary unit obtains the running configuration from the primary unit.

When the replication starts, the security appliance console on the unit sending the configuration displays the message “Beginning configuration replication: Sending to mate,” and when it is complete, the security appliance displays the message “End Configuration Replication to mate.” During replication, commands entered on the unit sending the configuration may not replicate properly to the peer unit, and commands entered on the unit receiving the configuration may be overwritten by the configuration being

received. Avoid entering commands on either unit in the failover pair during the configuration replication process. Depending upon the size of the configuration, replication can take from a few seconds to several minutes.

On the unit receiving the configuration, the configuration exists only in running memory. To save the configuration to Flash memory after synchronization enter the **write memory all** command in the system execution space on the unit that has failover group 1 in the active state. The command is replicated to the peer unit, which proceeds to write its configuration to Flash memory. Using the **all** keyword with this command causes the system and all context configurations to be saved.

Note Startup configurations saved on external servers are accessible from either unit over the network and do not need to be saved separately for each unit. Alternatively, you can copy the contexts configuration files from the disk on the primary unit to an external server, and then copy them to disk on the secondary unit, where they become available when the unit reloads.

Command Replication

After both units are running, commands are replicated from one unit to the other as follows:

- Commands entered within a security context are replicated from the unit on which the security context appears in the active state to the peer unit.

Note A context is considered in the active state on a unit if the failover group to which it belongs is in the active state on that unit.

- Commands entered in the system execution space are replicated from the unit on which failover group 1 is in the active state to the unit on which failover group 1 is in the standby state.
- Commands entered in the admin context are replicated from the unit on which failover group 1 is in the active state to the unit on which failover group 1 is in the standby state.

Failure to enter the commands on the appropriate unit for command replication to occur causes the configurations to be out of synchronization. Those changes may be lost the next time the initial configuration synchronization occurs.

Table 15-3 shows the commands that are and are not replicated to the standby unit:

Table 15-3 Command Replication

Commands Replicated to the Standby Unit	Commands Not Replicated to the Standby Unit
all configuration commands except for the **mode**, **firewall**, and **failover lan unit** commands	all forms of the **copy** command except for **copy running-config startup-config**
copy running-config startup-config	all forms of the **write** command except for **write memory**
delete	**debug**
mkdir	**failover lan unit**
rename	**firewall**
rmdir	**mode**
write memory	**show**

You can use the **write standby** command to resynchronize configurations that have become out of sync. For Active/Active failover, the **write standby** command behaves as follows:

- If you enter the **write standby** command in the system execution space, the system configuration and the configurations for all of the security contexts on the security appliance is written to the peer unit. This includes configuration information for security contexts that are in the standby state. You must enter the command in the system execution space on the unit that has failover group 1 in the active state.

  Note If there are security contexts in the active state on the peer unit, the **write standby** command causes active connections through those contexts to be terminated. Use the **failover active** command on the unit providing the configuration to make sure all contexts are active on that unit before entering the **write standby** command.

- If you enter the **write standby** command in a security context, only the configuration for the security context is written to the peer unit. You must enter the command in the security context on the unit where the security context appears in the active state.

Replicated commands are not saved to the Flash memory when replicated to the peer unit. They are added to the running configuration. To save replicated commands to Flash memory on both units, use the **write memory** or **copy running-config startup-config** command on the unit that you made the changes on. The command is replicated to the peer unit and cause the configuration to be saved to Flash memory on the peer unit.

Failover Triggers

In Active/Active failover, failover can be triggered at the unit level if one of the following events occurs:

- The unit has a hardware failure.
- The unit has a power failure.
- The unit has a software failure.
- The **no failover active** or the **failover active** command is entered in the system execution space.

Failover is triggered at the failover group level when one of the following events occurs:

- Too many monitored interfaces in the group fail.
- The **no failover active group** *group_id* or **failover active group** *group_id* command is entered.

You configure the failover threshold for each failover group by specifying the number or percentage of interfaces within the failover group that must fail before the group fails. Because a failover group can contain multiple contexts, and each context can contain multiple interfaces, it is possible for all interfaces in a single context to fail without causing the associated failover group to fail.

See the "Failover Health Monitoring" section on page 15-18 for more information about interface and unit monitoring.

Failover Actions

In an Active/Active failover configuration, failover occurs on a failover group basis, not a system basis. For example, if you designate both failover groups as active on the primary unit, and failover group 1 fails, then failover group 2 remains active on the primary unit while failover group 1 becomes active on the secondary unit.

Note When configuring Active/Active failover, make sure that the combined traffic for both units is within the capacity of each unit.

Table 15-4 shows the failover action for each failure event. For each failure event, the policy (whether or not failover occurs), actions for the active failover group, and actions for the standby failover group are given.

Table 15-4 Failover Behavior for Active/Active Failover

Failure Event	Policy	Active Group Action	Standby Group Action	Notes
A unit experiences a power or software failure	Failover	Become standby Mark as failed	Become active Mark active as failed	When a unit in a failover pair fails, any active failover groups on that unit are marked as failed and become active on the peer unit.
Interface failure on active failover group above threshold	Failover	Mark active group as failed	Become active	None.
Interface failure on standby failover group above threshold	No failover	No action	Mark standby group as failed	When the standby failover group is marked as failed, the active failover group does not attempt to fail over, even if the interface failure threshold is surpassed.
Formerly active failover group recovers	No failover	No action	No action	Unless configured with the **preempt** command, the failover groups remain active on their current unit.
Failover link failed at startup	No failover	Become active	Become active	If the failover link is down at startup, both failover groups on both units become active.
Stateful Failover link failed	No failover	No action	No action	State information becomes out of date, and sessions are terminated if a failover occurs.
Failover link failed during operation	No failover	n/a	n/a	Each unit marks the failover interface as failed. You should restore the failover link as soon as possible because the unit cannot fail over to the standby unit while the failover link is down.

Determining Which Type of Failover to Use

The type of failover you choose depends upon your security appliance configuration and how you plan to use the security appliances.

If you are running the security appliance in single mode, then you can use only Active/Standby failover. Active/Active failover is only available to security appliances running in multiple context mode.

If you are running the security appliance in multiple context mode, then you can configure either Active/Active failover or Active/Standby failover.

- To allow both members of the failover pair to share the traffic, use Active/Active failover. Do not exceed 50% load on each device.
- If you do not want to share the traffic in this way, use Active/Standby or Active/Active failover.

Table 15-5 provides a comparison of some of the features supported by each type of failover configuration:

Table 15-5 Failover Configuration Feature Support

Feature	Active/Active	Active/Standby
Single Context Mode	No	Yes
Multiple Context Mode	Yes	Yes
Traffic Sharing Network Configurations	Yes	No
Unit Failover	Yes	Yes
Failover of Groups of Contexts	Yes	No
Failover of Individual Contexts	No	No

Stateless (Regular) and Stateful Failover

The security appliance supports two types of failover, regular and stateful. This section includes the following topics:

- Stateless (Regular) Failover, page 15-16
- Stateful Failover, page 15-16

Stateless (Regular) Failover

When a failover occurs, all active connections are dropped. Clients need to reestablish connections when the new active unit takes over.

Note In Release 8.0 and later, some configuration elements for WebVPN (such as bookmarks and customization) use the VPN failover subsystem, which is part of Stateful Failover. You must use Stateful Failover to synchronize these elements between the members of the failover pair. Stateless (regular) failover is not recommended for WebVPN.

Stateful Failover

When Stateful Failover is enabled, the active unit continually passes per-connection state information to the standby unit. After a failover occurs, the same connection information is available at the new active unit. Supported end-user applications are not required to reconnect to keep the same communication session.

Table 15-6 list the state information that is and is not passed to the standby unit when Stateful Failover is enabled.

Table 15-6 State Information

State Information Passed to Standby Unit	State Information Not Passed to Standby Unit
NAT translation table.	The HTTP connection table (unless HTTP replication is enabled).
TCP connection states.	The user authentication (uauth) table.
UDP connection states.	The routing tables. After a failover occurs, some packets may be lost or routed out of the wrong interface (the default route) while the dynamic routing protocols rediscover routes.
The ARP table.	State information for Security Service Modules.
The Layer 2 bridge table (when running in transparent firewall mode).	DHCP server address leases.
The HTTP connection states (if HTTP replication is enabled).	Stateful failover for phone proxy. When the active unit goes down, the call fails, media stops flowing, and the phone should unregister from the failed unit and reregister with the active unit. The call must be re-established.
The ISAKMP and IPSec SA table.	—
GTP PDP connection database.	—
SIP signalling sessions.	—

The following WebVPN features are not supported with Stateful Failover:

- Smart Tunnels
- Port Forwarding
- Plugins
- Java Applets
- IPv6 clientless or Anyconnect sessions
- Citrix authentication (Citrix users must reauthenticate after failover)

Note If failover occurs during an active Cisco IP SoftPhone session, the call remains active because the call session state information is replicated to the standby unit. When the call is terminated, the IP SoftPhone client loses connection with the Cisco CallManager. This occurs because there is no session information for the CTIQBE hangup message on the standby unit. When the IP SoftPhone client does not receive a response back from the Call Manager within a certain time period, it considers the CallManager unreachable and unregisters itself.

For VPN failover, VPN end-users should not have to reauthenticate or reconnect the VPN session in the event of a failover. However, applications operating over the VPN connection could lose packets during the failover process and not recover from the packet loss.

Failover Health Monitoring

The security appliance monitors each unit for overall health and for interface health. See the following sections for more information about how the security appliance performs tests to determine the state of each unit:

- Unit Health Monitoring, page 15-18
- Interface Monitoring, page 15-18

Unit Health Monitoring

The security appliance determines the health of the other unit by monitoring the failover link. When a unit does not receive three consecutive hello messages on the failover link, the unit sends interface hello messages on each interface, including the failover interface, to validate whether or not the peer interface is responsive. The action that the security appliance takes depends upon the response from the other unit. See the following possible actions:

- If the security appliance receives a response on the failover interface, then it does not fail over.
- If the security appliance does not receive a response on the failover link, but receives a response on another interface, then the unit does not failover. The failover link is marked as failed. You should restore the failover link as soon as possible because the unit cannot fail over to the standby while the failover link is down.
- If the security appliance does not receive a response on any interface, then the standby unit switches to active mode and classifies the other unit as failed.

Note If a failed unit does not recover and you believe it should not be failed, you can reset the state by entering the **failover reset** command. If the failover condition persists, however, the unit will fail again.

You can configure the frequency of the hello messages and the hold time before failover occurs. A faster poll time and shorter hold time speed the detection of unit failures and make failover occur more quickly, but it can also cause “false” failures due to network congestion delaying the keepalive packets. See Configuring Unit Health Monitoring, page 15-41 for more information about configuring unit health monitoring.

Interface Monitoring

You can monitor up to 250 interfaces divided between all contexts. You should monitor important interfaces, for example, you might configure one context to monitor a shared interface (because the interface is shared, all contexts benefit from the monitoring).

When a unit does not receive hello messages on a monitored interface for half of the configured hold time, it runs the following tests:

1. Link Up/Down test—A test of the interface status. If the Link Up/Down test indicates that the interface is operational, then the security appliance performs network tests. The purpose of these tests is to generate network traffic to determine which (if either) unit has failed. At the start of each test, each unit clears its received packet count for its interfaces. At the conclusion of each test, each unit looks to see if it has received any traffic. If it has, the interface is considered operational. If one unit receives traffic for a test and the other unit does not, the unit that received no traffic is considered failed. If neither unit has received traffic, then the next test is used.

2. Network Activity test—A received network activity test. The unit counts all received packets for up to 5 seconds. If any packets are received at any time during this interval, the interface is considered operational and testing stops. If no traffic is received, the ARP test begins.
3. ARP test—A reading of the unit ARP cache for the 2 most recently acquired entries. One at a time, the unit sends ARP requests to these machines, attempting to stimulate network traffic. After each request, the unit counts all received traffic for up to 5 seconds. If traffic is received, the interface is considered operational. If no traffic is received, an ARP request is sent to the next machine. If at the end of the list no traffic has been received, the ping test begins.
4. Broadcast Ping test—A ping test that consists of sending out a broadcast ping request. The unit then counts all received packets for up to 5 seconds. If any packets are received at any time during this interval, the interface is considered operational and testing stops.

If all network tests fail for an interface, but this interface on the other unit continues to successfully pass traffic, then the interface is considered to be failed. If the threshold for failed interfaces is met, then a failover occurs. If the other unit interface also fails all the network tests, then both interfaces go into the "Unknown" state and do not count towards the failover limit.

An interface becomes operational again if it receives any traffic. A failed security appliance returns to standby mode if the interface failure threshold is no longer met.

Note If a failed unit does not recover and you believe it should not be failed, you can reset the state by entering the **failover reset** command. If the failover condition persists, however, the unit will fail again.

Failover Feature/Platform Matrix

Table 15-7 shows the failover features supported by each hardware platform.

Table 15-7 Failover Feature Support by Platform

Platform	Cable-Based Failover	LAN-Based Failover	Stateful Failover	Active/Standby Failover	Active/Active Failover
ASA 5505 adaptive security appliance	No	Yes	No	Yes	No
ASA 5500 series adaptive security appliance (other than the ASA 5505)	No	Yes	Yes	Yes	Yes
PIX 500 series security appliance	Yes	Yes	Yes	Yes	Yes

Failover Times by Platform

Table 15-8 shows the minimum, default, and maximum failover times for the PIX 500 series security appliance.

Table 15-8 PIX 500 Series Security Appliance Failover Times.

Failover Condition	Minimum	Default	Maximum
Active unit loses power or stops normal operation.	800 milliseconds	45 seconds	45 seconds
Active unit interface link down.	500 milliseconds	5 seconds	15 seconds
Active unit interface up, but connection problem causes interface testing.	5 seconds	25 seconds	75 seconds

Table 15-9 shows the minimum, default, and maximum failover times for the ASA 5500 series adaptive security appliance.

Table 15-9 ASA 5500 series adaptive security appliance failover times.

Failover Condition	Minimum	Default	Maximum
Active unit loses power or stops normal operation.	800 milliseconds	15 seconds	45 seconds
Active unit main board interface link down.	500 milliseconds	5 seconds	15 seconds
Active unit 4GE card interface link down.	2 seconds	5 seconds	15 seconds
Active unit IPS or CSC card fails.	2 seconds	2 seconds	2 seconds
Active unit interface up, but connection problem causes interface testing.	5 seconds	25 seconds	75 seconds

Configuring Failover

This section describes how to configure failover and includes the following topics:

- Failover Configuration Limitations, page 15-20
- Configuring Active/Standby Failover, page 15-21
- Configuring Active/Active Failover, page 15-29
- Configuring Unit Health Monitoring, page 15-41
- Configuring Failover Communication Authentication/Encryption, page 15-41
- Verifying the Failover Configuration, page 15-42

Failover Configuration Limitations

You cannot configure failover with the following type of IP addresses:

- IP addresses obtained through DHCP
- IP addresses obtained through PPPoE
- IPv6 addresses

Additionally, the following restrictions apply:

- Stateful Failover is not supported on the ASA 5505 adaptive security appliance.
- Active/Active failover is not supported on the ASA 5505 adaptive security appliance.
- You cannot configure failover when Easy VPN remote is enabled on the ASA 5505 adaptive security appliance.
- VPN failover is not supported in multiple context mode.
- CA server is not supported. If you have a CA server configured on the active unit, the CA server functionality will be lost when the unit fails over. The **crypto ca server** command and associated commands are not synchronized or replicated to the peer unit.

Configuring Active/Standby Failover

This section provides step-by-step procedures for configuring Active/Standby failover. This section includes the following topics:

- Prerequisites, page 15-21
- Configuring Cable-Based Active/Standby Failover (PIX 500 Series Security Appliance Only), page 15-21
- Configuring LAN-Based Active/Standby Failover, page 15-23
- Configuring Optional Active/Standby Failover Settings, page 15-26

Prerequisites

Before you begin, verify the following:

- Both units have the same hardware, software configuration, and proper license.
- Both units are in the same mode (single or multiple, transparent or routed).

Configuring Cable-Based Active/Standby Failover (PIX 500 Series Security Appliance Only)

Follow these steps to configure Active/Standby failover using a serial cable as the failover link. The commands in this task are entered on the *primary* unit in the failover pair. The primary unit is the unit that has the end of the cable labeled “Primary” plugged into it. For devices in multiple context mode, the commands are entered in the system execution space unless otherwise noted.

You do not need to bootstrap the secondary unit in the failover pair when you use cable-based failover. Leave the secondary unit powered off until instructed to power it on.

Cable-based failover is only available on the PIX 500 series security appliance.

To configure cable-based Active/Standby failover, perform the following steps:

Step 1 Connect the Failover cable to the PIX 500 series security appliances. Make sure that you attach the end of the cable marked “Primary” to the unit you use as the primary unit, and that you attach the end of the cable marked “Secondary” to the other unit.

Step 2 Power on the primary unit.

Step 3 If you have not done so already, configure the active and standby IP addresses for each data interface (routed mode), for the management IP address (transparent mode), or for the management-only interface. To receive packets from both units in a failover pair, standby IP addresses need to be configured on all interfaces. The standby IP address is used on the security appliance that is currently the standby unit, and it must be in the same subnet as the active IP address.

Note Do not configure an IP address for the Stateful Failover link if you are going to use a dedicated Stateful Failover interface. You use the **failover interface ip** command to configure a dedicated Stateful Failover interface in a later step.

```
hostname(config-if)# ip address active_addr netmask standby standby_addr
```

In routed firewall mode and for the management-only interface, this command is entered in interface configuration mode for each interface. In transparent firewall mode, the command is entered in global configuration mode.

In multiple context mode, you must configure the interface addresses from within each context. Use the **changeto context** command to switch between contexts. The command prompt changes to `hostname/context(config-if)#`, where *context* is the name of the current context. You must enter a management IP address for each context in transparent firewall multiple context mode.

Step 4 (Optional) To enable Stateful Failover, configure the Stateful Failover link.

Note Stateful Failover is not available on the ASA 5505 adaptive security appliance.

a. Specify the interface to be used as the Stateful Failover link:

```
hostname(config)# failover link if_name phy_if
```

The *if_name* argument assigns a logical name to the interface specified by the *phy_if* argument. The *phy_if* argument can be the physical port name, such as Ethernet1, or a previously created subinterface, such as Ethernet0/2.3. This interface should not be used for any other purpose.

b. Assign an active and standby IP address to the Stateful Failover link:

```
hostname(config)# failover interface ip if_name ip_addr mask standby ip_addr
```

Note If the Stateful Failover link uses a data interface, skip this step. You have already defined the active and standby IP addresses for the interface.

The standby IP address must be in the same subnet as the active IP address. You do not need to identify the standby IP address subnet mask.

The Stateful Failover link IP address and MAC address do not change at failover unless it uses a data interface. The active IP address always stays with the primary unit, while the standby IP address stays with the secondary unit.

c. Enable the interface:

```
hostname(config)# interface phy_if
hostname(config-if)# no shutdown
```

Step 5 Enable failover:

```
hostname(config)# failover
```

Step 6 Power on the secondary unit and enable failover on the unit if it is not already enabled:

```
hostname(config)# failover
```

The active unit sends the configuration in running memory to the standby unit. As the configuration synchronizes, the messages "Beginning configuration replication: sending to mate." and "End Configuration Replication to mate" appear on the primary console.

Step 7 Save the configuration to Flash memory on the primary unit. Because the commands entered on the primary unit are replicated to the secondary unit, the secondary unit also saves its configuration to Flash memory.

```
hostname(config)# copy running-config startup-config
```

Configuring LAN-Based Active/Standby Failover

This section describes how to configure Active/Standby failover using an Ethernet failover link. When configuring LAN-based failover, you must bootstrap the secondary device to recognize the failover link before the secondary device can obtain the running configuration from the primary device.

Note If you are changing from cable-based failover to LAN-based failover, you can skip any steps, such as assigning the active and standby IP addresses for each interface, that you completed for the cable-based failover configuration.

This section includes the following topics:

- Configuring the Primary Unit, page 15-23
- Configuring the Secondary Unit, page 15-25

Configuring the Primary Unit

Follow these steps to configure the primary unit in a LAN-based, Active/Standby failover configuration. These steps provide the minimum configuration needed to enable failover on the primary unit. For multiple context mode, all steps are performed in the system execution space unless otherwise noted.

To configure the primary unit in an Active/Standby failover pair, perform the following steps:

Step 1 If you have not done so already, configure the active and standby IP addresses for each data interface (routed mode), for the management IP address (transparent mode), or for the management-only interface. To receive packets from both units in a failover pair, standby IP addresses need to be configured on all interfaces. The standby IP address is used on the security appliance that is currently the standby unit, and it must be in the same subnet as the active IP address.

Note Do not configure an IP address for the Stateful Failover link if you are going to use a dedicated Stateful Failover interface. You use the **failover interface ip** command to configure a dedicated Stateful Failover interface in a later step.

```
hostname(config-if)# ip address active_addr netmask standby standby_addr
```

In routed firewall mode and for the management-only interface, this command is entered in interface configuration mode for each interface. In transparent firewall mode, the command is entered in global configuration mode.

In multiple context mode, you must configure the interface addresses from within each context. Use the **changeto context** command to switch between contexts. The command prompt changes to `hostname/context(config-if)#`, where *context* is the name of the current context. You must enter a management IP address for each context in transparent firewall multiple context mode.

Step 2 (PIX 500 series security appliance only) Enable LAN-based failover:

```
hostname(config)# failover lan enable
```

Step 3 Designate the unit as the primary unit:

```
hostname(config)# failover lan unit primary
```

Step 4 Define the failover interface:

a. Specify the interface to be used as the failover interface:

```
hostname(config)# failover lan interface if_name phy_if
```

The *if_name* argument assigns a name to the interface specified by the *phy_if* argument. The *phy_if* argument can be the physical port name, such as Ethernet1, or a previously created subinterface, such as Ethernet0/2.3. On the ASA 5505 adaptive security appliance, the *phy_if* specifies a VLAN.

b. Assign the active and standby IP address to the failover link:

```
hostname(config)# failover interface ip if_name ip_addr mask standby ip_addr
```

The standby IP address must be in the same subnet as the active IP address. You do not need to identify the standby address subnet mask.

The failover link IP address and MAC address do not change at failover. The active IP address for the failover link always stays with the primary unit, while the standby IP address stays with the secondary unit.

c. Enable the interface:

```
hostname(config)# interface phy_if
hostname(config-if)# no shutdown
```

Step 5 (Optional) To enable Stateful Failover, configure the Stateful Failover link.

Note Stateful Failover is not available on the ASA 5505 adaptive security appliance.

a. Specify the interface to be used as Stateful Failover link:

```
hostname(config)# failover link if_name phy_if
```

Note If the Stateful Failover link uses the failover link or a data interface, then you only need to supply the *if_name* argument.

The *if_name* argument assigns a logical name to the interface specified by the *phy_if* argument. The *phy_if* argument can be the physical port name, such as Ethernet1, or a previously created subinterface, such as Ethernet0/2.3. This interface should not be used for any other purpose (except, optionally, the failover link).

b. Assign an active and standby IP address to the Stateful Failover link.

Note If the Stateful Failover link uses the failover link or data interface, skip this step. You have already defined the active and standby IP addresses for the interface.

```
hostname(config)# failover interface ip if_name ip_addr mask standby ip_addr
```

The standby IP address must be in the same subnet as the active IP address. You do not need to identify the standby address subnet mask.

The Stateful Failover link IP address and MAC address do not change at failover unless it uses a data interface. The active IP address always stays with the primary unit, while the standby IP address stays with the secondary unit.

c. Enable the interface.

Note If the Stateful Failover link uses the failover link or data interface, skip this step. You have already enabled the interface.

```
hostname(config)# interface phy_if
hostname(config-if)# no shutdown
```

Step 6 Enable failover:

```
hostname(config)# failover
```

Step 7 Save the system configuration to Flash memory:

```
hostname(config)# copy running-config startup-config
```

Configuring the Secondary Unit

The only configuration required on the secondary unit is for the failover interface. The secondary unit requires these commands to initially communicate with the primary unit. After the primary unit sends its configuration to the secondary unit, the only permanent difference between the two configurations is the **failover lan unit** command, which identifies each unit as primary or secondary.

For multiple context mode, all steps are performed in the system execution space unless noted otherwise.

To configure the secondary unit, perform the following steps:

Step 1 (PIX 500 series security appliance only) Enable LAN-based failover:

```
hostname(config)# failover lan enable
```

Step 2 Define the failover interface. Use the same settings as you used for the primary unit.

a. Specify the interface to be used as the failover interface:

```
hostname(config)# failover lan interface if_name phy_if
```

The *if_name* argument assigns a name to the interface specified by the *phy_if* argument.

b. Assign the active and standby IP address to the failover link. To receive packets from both units in a failover pair, standby IP addresses need to be configured on all interfaces.

```
hostname(config)# failover interface ip if_name ip_addr mask standby ip_addr
```

Note Enter this command exactly as you entered it on the primary unit when you configured the failover interface on the primary unit (including the same IP address).

c. Enable the interface:

```
hostname(config)# interface phy_if
hostname(config-if)# no shutdown
```

Step 3 (Optional) Designate this unit as the secondary unit:

```
hostname(config)# failover lan unit secondary
```

Note This step is optional because by default units are designated as secondary unless previously configured.

Step 4 Enable failover:

```
hostname(config)# failover
```

After you enable failover, the active unit sends the configuration in running memory to the standby unit. As the configuration synchronizes, the messages "Beginning configuration replication: Sending to mate" and "End Configuration Replication to mate" appear on the active unit console.

Step 5 After the running configuration has completed replication, save the configuration to Flash memory:

```
hostname(config)# copy running-config startup-config
```

Configuring Optional Active/Standby Failover Settings

You can configure the following optional Active/Standby failover setting when you are initially configuring failover or after failover has already been configured. Unless otherwise noted, the commands should be entered on the active unit.

This section includes the following topics:

- Enabling HTTP Replication with Stateful Failover, page 15-26
- Disabling and Enabling Interface Monitoring, page 15-27
- Configuring Interface Health Monitoring, page 15-27
- Configuring Failover Criteria, page 15-28
- Configuring Virtual MAC Addresses, page 15-28

Enabling HTTP Replication with Stateful Failover

To allow HTTP connections to be included in the state information replication, you need to enable HTTP replication. Because HTTP connections are typically short-lived, and because HTTP clients typically retry failed connection attempts, HTTP connections are not automatically included in the replicated state information.

Enter the following command in global configuration mode to enable HTTP state replication when Stateful Failover is enabled:

```
hostname(config)# failover replication http
```

Disabling and Enabling Interface Monitoring

By default, monitoring physical interfaces is enabled and monitoring subinterfaces is disabled. You can monitor up to 250 interfaces on a unit. You can control which interfaces affect your failover policy by disabling the monitoring of specific interfaces and enabling the monitoring of others. This lets you exclude interfaces attached to less critical networks from affecting your failover policy.

For units in multiple configuration mode, use the following commands to enable or disable health monitoring for specific interfaces:

- To disable health monitoring for an interface, enter the following command within a context:

  ```
  hostname/context(config)# no monitor-interface if_name
  ```

- To enable health monitoring for an interface, enter the following command within a context:

  ```
  hostname/context(config)# monitor-interface if_name
  ```

For units in single configuration mode, use the following commands to enable or disable health monitoring for specific interfaces:

- To disable health monitoring for an interface, enter the following command in global configuration mode:

  ```
  hostname(config)# no monitor-interface if_name
  ```

- To enable health monitoring for an interface, enter the following command in global configuration mode:

  ```
  hostname(config)# monitor-interface if_name
  ```

Configuring Interface Health Monitoring

The security appliance sends hello packets out of each data interface to monitor interface health. If the security appliance does not receive a hello packet from the corresponding interface on the peer unit for over half of the hold time, then the additional interface testing begins. If a hello packet or a successful test result is not received within the specified hold time, the interface is marked as failed. Failover occurs if the number of failed interfaces meets the failover criteria.

Decreasing the poll and hold times enables the security appliance to detect and respond to interface failures more quickly, but may consume more system resources.

To change the interface poll time, enter the following command in global configuration mode:

```
hostname(config)# failover polltime interface [msec] time [holdtime time]
```

Valid values for the poll time are from 1 to 15 seconds or, if the optional msec keyword is used, from 500 to 999 milliseconds. The hold time determines how long it takes from the time a hello packet is missed to when the interface is marked as failed. Valid values for the hold time are from 5 to 75 seconds. You cannot enter a hold time that is less than 5 times the poll time.



Note If the interface link is down, interface testing is not conducted and the standby unit could become active in just one interface polling period if the number of failed interface meets or exceeds the configured failover criteria.

Configuring Failover Criteria

By default, a single interface failure causes failover. You can specify a specific number of interfaces or a percentage of monitored interfaces that must fail before a failover occurs.

To change the default failover criteria, enter the following command in global configuration mode:

```
hostname(config)# failover interface-policy num[%]
```

When specifying a specific number of interfaces, the *num* argument can be from 1 to 250. When specifying a percentage of interfaces, the *num* argument can be from 1 to 100.

Configuring Virtual MAC Addresses

In Active/Standby failover, the MAC addresses for the primary unit are always associated with the active IP addresses. If the secondary unit boots first and becomes active, it uses the burned-in MAC address for its interfaces. When the primary unit comes online, the secondary unit obtains the MAC addresses from the primary unit. The change can disrupt network traffic.

You can configure virtual MAC addresses for each interface to ensure that the secondary unit uses the correct MAC addresses when it is the active unit, even if it comes online before the primary unit. If you do not specify virtual MAC addresses the failover pair uses the burned-in NIC addresses as the MAC addresses.

Note You cannot configure a virtual MAC address for the failover or Stateful Failover links. The MAC and IP addresses for those links do not change during failover.

Enter the following command on the active unit to configure the virtual MAC addresses for an interface:

```
hostname(config)# failover mac address phy_if active_mac standby_mac
```

The *phy_if* argument is the physical name of the interface, such as Ethernet1. The *active_mac* and *standby_mac* arguments are MAC addresses in H.H.H format, where H is a 16-bit hexadecimal digit. For example, the MAC address 00-0C-F1-42-4C-DE would be entered as 000C.F142.4CDE.

The *active_mac* address is associated with the active IP address for the interface, and the *standby_mac* is associated with the standby IP address for the interface.

There are multiple ways to configure virtual MAC addresses on the security appliance. When more than one method has been used to configure virtual MAC addresses, the security appliance uses the following order of preference to determine which virtual MAC address is assigned to an interface:

1. The **mac-address** command (in interface configuration mode) address.
2. The **failover mac address** command address.
3. The **mac-address auto** command generated address.
4. The burned-in MAC address.

Use the **show interface** command to display the MAC address used by an interface.

Configuring Active/Active Failover

This section describes how to configure Active/Active failover.

Note Active/Active failover is not available on the ASA 5505 adaptive security appliance.

This section includes the following topics:

- Prerequisites, page 15-29
- Configuring Cable-Based Active/Active Failover (PIX 500 series security appliance), page 15-29
- Configuring LAN-Based Active/Active Failover, page 15-31
- Configuring Optional Active/Active Failover Settings, page 15-35

Prerequisites

Before you begin, verify the following:

- Both units have the same hardware, software configuration, and proper license.
- Both units are in multiple context mode.

Configuring Cable-Based Active/Active Failover (PIX 500 series security appliance)

Follow these steps to configure Active/Active failover using a serial cable as the failover link. The commands in this task are entered on the *primary* unit in the failover pair. The primary unit is the unit that has the end of the cable labeled "Primary" plugged into it. For devices in multiple context mode, the commands are entered in the system execution space unless otherwise noted.

You do not need to bootstrap the secondary unit in the failover pair when you use cable-based failover. Leave the secondary unit powered off until instructed to power it on.

Cable-based failover is only available on the PIX 500 series security appliance.

To configure cable-based, Active/Active failover, perform the following steps:

Step 1 Connect the failover cable to the PIX 500 series security appliances. Make sure that you attach the end of the cable marked "Primary" to the unit you use as the primary unit, and that you attach the end of the cable marked "Secondary" to the unit you use as the secondary unit.

Step 2 Power on the primary unit.

Step 3 If you have not done so already, configure the active and standby IP addresses for each data interface (routed mode), for the management IP address (transparent mode), or for the management-only interface. To receive packets from both units in a failover pair, standby IP addresses need to be configured on all interfaces. The standby IP address is used on the security appliance that is currently the standby unit, and it must be in the same subnet as the active IP address.

You must configure the interface addresses from within each context. Use the **changeto context** command to switch between contexts. The command prompt changes to `hostname/context(config-if)#`, where *context* is the name of the current context. You must enter a management IP address for each context in transparent firewall multiple context mode.

Note Do not configure an IP address for the Stateful Failover link if you are going to use a dedicated Stateful Failover interface. You use the **failover interface ip** command to configure a dedicated Stateful Failover interface in a later step.

```
hostname/context(config-if)# ip address active_addr netmask standby standby_addr
```

In routed firewall mode and for the management-only interface, this command is entered in interface configuration mode for each interface. In transparent firewall mode, the command is entered in global configuration mode.

Step 4 (Optional) To enable Stateful Failover, configure the Stateful Failover link.

a. Specify the interface to be used as Stateful Failover link:

```
hostname(config)# failover link if_name phy_if
```

The *if_name* argument assigns a logical name to the interface specified by the *phy_if* argument. The *phy_if* argument can be the physical port name, such as Ethernet1, or a previously created subinterface, such as Ethernet0/2.3. This interface should not be used for any other purpose (except, optionally, the failover link).

b. Assign an active and standby IP address to the Stateful Failover link:

```
hostname(config)# failover interface ip if_name ip_addr mask standby ip_addr
```

The standby IP address must be in the same subnet as the active IP address. You do not need to identify the standby IP address subnet mask.

The Stateful Failover link IP address and MAC address do not change at failover except for when Stateful Failover uses a regular data interface. The active IP address always stays with the primary unit, while the standby IP address stays with the secondary unit.

c. Enable the interface:

```
hostname(config)# interface phy_if
hostname(config-if)# no shutdown
```

Step 5 Configure the failover groups. You can have at most two failover groups. The **failover group** command creates the specified failover group if it does not exist and enters the failover group configuration mode.

For each failover group, you need to specify whether the failover group has primary or secondary preference using the **primary** or **secondary** command. You can assign the same preference to both failover groups. For traffic sharing configurations, you should assign each failover group a different unit preference.

The following example assigns failover group 1 a primary preference and failover group 2 a secondary preference:

```
hostname(config)# failover group 1
hostname(config-fover-group)# primary
hostname(config-fover-group)# exit
hostname(config)# failover group 2
hostname(config-fover-group)# secondary
hostname(config-fover-group)# exit
```

Step 6 Assign each user context to a failover group using the **join-failover-group** command in context configuration mode.

Any unassigned contexts are automatically assigned to failover group 1. The admin context is always a member of failover group 1.

Enter the following commands to assign each context to a failover group:

```
hostname(config)# context context_name
hostname(config-context)# join-failover-group {1 | 2}
hostname(config-context)# exit
```

Step 7 Enable failover:

```
hostname(config)# failover
```

Step 8 Power on the secondary unit and enable failover on the unit if it is not already enabled:

```
hostname(config)# failover
```

The active unit sends the configuration in running memory to the standby unit. As the configuration synchronizes, the messages "Beginning configuration replication: Sending to mate" and "End Configuration Replication to mate" appear on the primary console.

Step 9 Save the configuration to Flash memory on the Primary unit. Because the commands entered on the primary unit are replicated to the secondary unit, the secondary unit also saves its configuration to Flash memory.

```
hostname(config)# copy running-config startup-config
```

Step 10 If necessary, force any failover group that is active on the primary to the active state on the secondary. To force a failover group to become active on the secondary unit, issue the following command in the system execution space on the primary unit:

```
hostname# no failover active group group_id
```

The *group_id* argument specifies the group you want to become active on the secondary unit.

Configuring LAN-Based Active/Active Failover

This section describes how to configure Active/Active failover using an Ethernet failover link. When configuring LAN-based failover, you must bootstrap the secondary device to recognize the failover link before the secondary device can obtain the running configuration from the primary device.

This section includes the following topics:

- Configure the Primary Unit, page 15-31
- Configure the Secondary Unit, page 15-33

Configure the Primary Unit

To configure the primary unit in an Active/Active failover configuration, perform the following steps:

Step 1 If you have not done so already, configure the active and standby IP addresses for each data interface (routed mode), for the management IP address (transparent mode), or for the management-only interface. To receive packets from both units in a failover pair, standby IP addresses need to be configured on all interfaces. The standby IP address is used on the security appliance that is currently the standby unit, and it must be in the same subnet as the active IP address.

You must configure the interface addresses from within each context. Use the **changeto context** command to switch between contexts. The command prompt changes to `hostname/context(config-if)#`, where *context* is the name of the current context. In transparent firewall mode, you must enter a management IP address for each context.

Note Do not configure an IP address for the Stateful Failover link if you are going to use a dedicated Stateful Failover interface. You use the **failover interface ip** command to configure a dedicated Stateful Failover interface in a later step.

```
hostname/context(config-if)# ip address active_addr netmask standby standby_addr
```

In routed firewall mode and for the management-only interface, this command is entered in interface configuration mode for each interface. In transparent firewall mode, the command is entered in global configuration mode.

Step 2 Configure the basic failover parameters in the system execution space.

a. (PIX 500 series security appliance only) Enable LAN-based failover:

```
hostname(config)# hostname(config)# failover lan enable
```

b. Designate the unit as the primary unit:

```
hostname(config)# failover lan unit primary
```

c. Specify the failover link:

```
hostname(config)# failover lan interface if_name phy_if
```

The *if_name* argument assigns a logical name to the interface specified by the *phy_if* argument. The *phy_if* argument can be the physical port name, such as Ethernet1, or a previously created subinterface, such as Ethernet0/2.3. On the ASA 5505 adaptive security appliance, the *phy_if* specifies a VLAN. This interface should not be used for any other purpose (except, optionally, the Stateful Failover link).

d. Specify the failover link active and standby IP addresses:

```
hostname(config)# failover interface ip if_name ip_addr mask standby ip_addr
```

The standby IP address must be in the same subnet as the active IP address. You do not need to identify the standby IP address subnet mask. The failover link IP address and MAC address do not change at failover. The active IP address always stays with the primary unit, while the standby IP address stays with the secondary unit.

Step 3 (Optional) To enable Stateful Failover, configure the Stateful Failover link:

a. Specify the interface to be used as Stateful Failover link:

```
hostname(config)# failover link if_name phy_if
```

The *if_name* argument assigns a logical name to the interface specified by the *phy_if* argument. The *phy_if* argument can be the physical port name, such as Ethernet1, or a previously created subinterface, such as Ethernet0/2.3. This interface should not be used for any other purpose (except, optionally, the failover link).

Note If the Stateful Failover link uses the failover link or a regular data interface, then you only need to supply the *if_name* argument.

b. Assign an active and standby IP address to the Stateful Failover link.

Note If the Stateful Failover link uses the failover link or a regular data interface, skip this step. You have already defined the active and standby IP addresses for the interface.

```
hostname(config)# failover interface ip if_name ip_addr mask standby ip_addr
```

The standby IP address must be in the same subnet as the active IP address. You do not need to identify the standby address subnet mask.

The state link IP address and MAC address do not change at failover. The active IP address always stays with the primary unit, while the standby IP address stays with the secondary unit.

c. Enable the interface.

Note If the Stateful Failover link uses the failover link or regular data interface, skip this step. You have already enabled the interface.

```
hostname(config)# interface phy_if
hostname(config-if)# no shutdown
```

Step 4 Configure the failover groups. You can have at most two failover groups. The **failover group** command creates the specified failover group if it does not exist and enters the failover group configuration mode.

For each failover group, specify whether the failover group has primary or secondary preference using the **primary** or **secondary** command. You can assign the same preference to both failover groups. For traffic sharing configurations, you should assign each failover group a different unit preference.

The following example assigns failover group 1 a primary preference and failover group 2 a secondary preference:

```
hostname(config)# failover group 1
hostname(config-fover-group)# primary
hostname(config-fover-group)# exit
hostname(config)# failover group 2
hostname(config-fover-group)# secondary
hostname(config-fover-group)# exit
```

Step 5 Assign each user context to a failover group using the **join-failover-group** command in context configuration mode.

Any unassigned contexts are automatically assigned to failover group 1. The admin context is always a member of failover group 1.

Enter the following commands to assign each context to a failover group:

```
hostname(config)# context context_name
hostname(config-context)# join-failover-group {1 | 2}
hostname(config-context)# exit
```

Step 6 Enable failover:

```
hostname(config)# failover
```

Configure the Secondary Unit

When configuring LAN-based Active/Active failover, you need to bootstrap the secondary unit to recognize the failover link. This allows the secondary unit to communicate with and receive the running configuration from the primary unit.

To bootstrap the secondary unit in an Active/Active failover configuration, perform the following steps:

Step 1 (PIX 500 series security appliance only) Enable LAN-based failover:

```
hostname(config)# failover lan enable
```

Step 2 Define the failover interface. Use the same settings as you used for the primary unit:

a. Specify the interface to be used as the failover interface:

```
hostname(config)# failover lan interface if_name phy_if
```

The *if_name* argument assigns a logical name to the interface specified by the *phy_if* argument. The *phy_if* argument can be the physical port name, such as Ethernet1, or a previously created subinterface, such as Ethernet0/2.3. On the ASA 5505 adaptive security appliance, the *phy_if* specifies a VLAN.

b. Assign the active and standby IP address to the failover link. To receive packets from both units in a failover pair, standby IP addresses need to be configured on all interfaces.

```
hostname(config)# failover interface ip if_name ip_addr mask standby ip_addr
```

Note Enter this command exactly as you entered it on the primary unit when you configured the failover interface (including the same IP address).

The standby IP address must be in the same subnet as the active IP address. You do not need to identify the standby address subnet mask.

c. Enable the interface:

```
hostname(config)# interface phy_if
hostname(config-if)# no shutdown
```

Step 3 (Optional) Designate this unit as the secondary unit:

```
hostname(config)# failover lan unit secondary
```

Note This step is optional because by default units are designated as secondary unless previously configured otherwise.

Step 4 Enable failover:

```
hostname(config)# failover
```

After you enable failover, the active unit sends the configuration in running memory to the standby unit. As the configuration synchronizes, the messages `Beginning configuration replication: Sending to mate` and `End Configuration Replication to mate` appear on the active unit console.

Step 5 After the running configuration has completed replication, enter the following command to save the configuration to Flash memory:

```
hostname(config)# copy running-config startup-config
```

Step 6 If necessary, force any failover group that is active on the primary to the active state on the secondary unit. To force a failover group to become active on the secondary unit, enter the following command in the system execution space on the primary unit:

```
hostname# no failover active group group_id
```

The *group_id* argument specifies the group you want to become active on the secondary unit.

Configuring Optional Active/Active Failover Settings

The following optional Active/Active failover settings can be configured when you are initially configuring failover or after you have already established failover. Unless otherwise noted, the commands should be entered on the unit that has failover group 1 in the active state.

This section includes the following topics:

- Configuring Failover Group Preemption, page 15-35
- Enabling HTTP Replication with Stateful Failover, page 15-35
- Disabling and Enabling Interface Monitoring, page 15-36
- Configuring Interface Health Monitoring, page 15-36
- Configuring Failover Criteria, page 15-36
- Configuring Virtual MAC Addresses, page 15-36
- Configuring Support for Asymmetrically Routed Packets, page 15-37

Configuring Failover Group Preemption

Assigning a primary or secondary priority to a failover group specifies which unit the failover group becomes active on when both units boot simultaneously. However, if one unit boots before the other, then both failover groups become active on that unit. When the other unit comes online, any failover groups that have the unit as a priority do not become active on that unit unless manually forced over, a failover occurs, or the failover group is configured with the **preempt** command. The **preempt** command causes a failover group to become active on the designated unit automatically when that unit becomes available.

Enter the following commands to configure preemption for the specified failover group:

```
hostname(config)# failover group {1 | 2}
hostname(config-fover-group)# preempt [delay]
```

You can enter an optional *delay* value, which specifies the number of seconds the failover group remains active on the current unit before automatically becoming active on the designated unit.

Enabling HTTP Replication with Stateful Failover

To allow HTTP connections to be included in the state information, you need to enable HTTP replication. Because HTTP connections are typically short-lived, and because HTTP clients typically retry failed connection attempts, HTTP connections are not automatically included in the replicated state information. You can use the **replication http** command to cause a failover group to replicate HTTP state information when Stateful Failover is enabled.

To enable HTTP state replication for a failover group, enter the following command. This command only affects the failover group in which it was configured. To enable HTTP state replication for both failover groups, you must enter this command in each group. This command should be entered in the system execution space.

```
hostname(config)# failover group {1 | 2}
hostname(config-fover-group)# replication http
```

Disabling and Enabling Interface Monitoring

You can monitor up to 250 interfaces on a unit. By default, monitoring of physical interfaces is enabled and the monitoring of subinterfaces is disabled. You can control which interfaces affect your failover policy by disabling the monitoring of specific interfaces and enabling the monitoring of others. This lets you exclude interfaces attached to less critical networks from affecting your failover policy.

To disable health monitoring on an interface, enter the following command within a context:

```
hostname/context(config)# no monitor-interface if_name
```

To enable health monitoring on an interface, enter the following command within a context:

```
hostname/context(config)# monitor-interface if_name
```

Configuring Interface Health Monitoring

The security appliance sends hello packets out of each data interface to monitor interface health. If the security appliance does not receive a hello packet from the corresponding interface on the peer unit for over half of the hold time, then the additional interface testing begins. If a hello packet or a successful test result is not received within the specified hold time, the interface is marked as failed. Failover occurs if the number of failed interfaces meets the failover criteria.

Decreasing the poll and hold times enables the security appliance to detect and respond to interface failures more quickly, but may consume more system resources.

To change the default interface poll time, enter the following commands:

```
hostname(config)# failover group {1 | 2}
hostname(config-fover-group)# polltime interface seconds
```

Valid values for the poll time are from 1 to 15 seconds or, if the optional msec keyword is used, from 500 to 999 milliseconds. The hold time determines how long it takes from the time a hello packet is missed to when the interface is marked as failed. Valid values for the hold time are from 5 to 75 seconds. You cannot enter a hold time that is less than 5 times the poll time.

Configuring Failover Criteria

By default, if a single interface fails failover occurs. You can specify a specific number of interfaces or a percentage of monitored interfaces that must fail before a failover occurs. The failover criteria is specified on a failover group basis.

To change the default failover criteria for the specified failover group, enter the following commands:

```
hostname(config)# failover group {1 | 2}
hostname(config-fover-group)# interface-policy num[%]
```

When specifying a specific number of interfaces, the *num* argument can be from 1 to 250. When specifying a percentage of interfaces, the *num* argument can be from 1 to 100.

Configuring Virtual MAC Addresses

Active/Active failover uses virtual MAC addresses on all interfaces. If you do not specify the virtual MAC addresses, then they are computed as follows:

- Active unit default MAC address: 00a0.c9*physical_port_number.failover_group_id*01.
- Standby unit default MAC address: 00a0.c9*physical_port_number.failover_group_id*02.

Note If you have more than one Active/Active failover pair on the same network, it is possible to have the same default virtual MAC addresses assigned to the interfaces on one pair as are assigned to the interfaces of the other pairs because of the way the default virtual MAC addresses are determined. To avoid having duplicate MAC addresses on your network, make sure you assign each physical interface a virtual active and standby MAC address for all failover groups.

You can configure specific active and standby MAC addresses for an interface by entering the following commands:

```
hostname(config)# failover group {1 | 2}
hostname(config-fover-group)# mac address phy_if active_mac standby_mac
```

The *phy_if* argument is the physical name of the interface, such as Ethernet1. The *active_mac* and *standby_mac* arguments are MAC addresses in H.H.H format, where H is a 16-bit hexadecimal digit. For example, the MAC address 00-0C-F1-42-4C-DE would be entered as 000C.F142.4CDE.

The *active_mac* address is associated with the active IP address for the interface, and the *standby_mac* is associated with the standby IP address for the interface.

There are multiple ways to configure virtual MAC addresses on the security appliance. When more than one method has been used to configure virtual MAC addresses, the security appliance uses the following order of preference to determine which virtual MAC address is assigned to an interface:

1. The **mac-address** command (in interface configuration mode) address.
2. The **failover mac address** command address.
3. The **mac-address auto** command generate address.
4. The automatically generated failover MAC address.

Use the **show interface** command to display the MAC address used by an interface.

Configuring Support for Asymmetrically Routed Packets

When running in Active/Active failover, a unit may receive a return packet for a connection that originated through its peer unit. Because the security appliance that receives the packet does not have any connection information for the packet, the packet is dropped. This most commonly occurs when the two security appliances in an Active/Active failover pair are connected to different service providers and the outbound connection does not use a NAT address.

You can prevent the return packets from being dropped using the **asr-group** command on interfaces where this is likely to occur. When an interface configured with the **asr-group** command receives a packet for which it has no session information, it checks the session information for the other interfaces that are in the same group. If it does not find a match, the packet is dropped. If it finds a match, then one of the following actions occurs:

- If the incoming traffic originated on a peer unit, some or all of the layer 2 header is rewritten and the packet is redirected to the other unit. This redirection continues as long as the session is active.
- If the incoming traffic originated on a different interface on the same unit, some or all of the layer 2 header is rewritten and the packet is reinjected into the stream.

Note Using the **asr-group** command to configure asymmetric routing support is more secure than using the **static** command with the **nailed** option.

The **asr-group** command does not provide asymmetric routing; it restores asymmetrically routed packets to the correct interface.

Prerequisites

You must have to following configured for asymmetric routing support to function properly:

- Active/Active Failover
- Stateful Failover—passes state information for sessions on interfaces in the active failover group to the standby failover group.
- **replication http**—HTTP session state information is not passed to the standby failover group, and therefore is not present on the standby interface. For the security appliance to be able re-route asymmetrically routed HTTP packets, you need to replicate the HTTP state information.

You can configure the **asr-group** command on an interface without having failover configured, but it does not have any effect until Stateful Failover is enabled.

Configuring Support for Asymmetrically Routed Packets

To configure support for asymmetrically routed packets, perform the following steps:

Step 1 Configure Active/Active Stateful Failover for the failover pair. See Configuring Active/Active Failover, page 15-29.

Step 2 For each interface that you want to participate in asymmetric routing support enter the following command. You must enter the command on the unit where the context is in the active state so that the command is replicated to the standby failover group. For more information about command replication, see Command Replication, page 15-13.

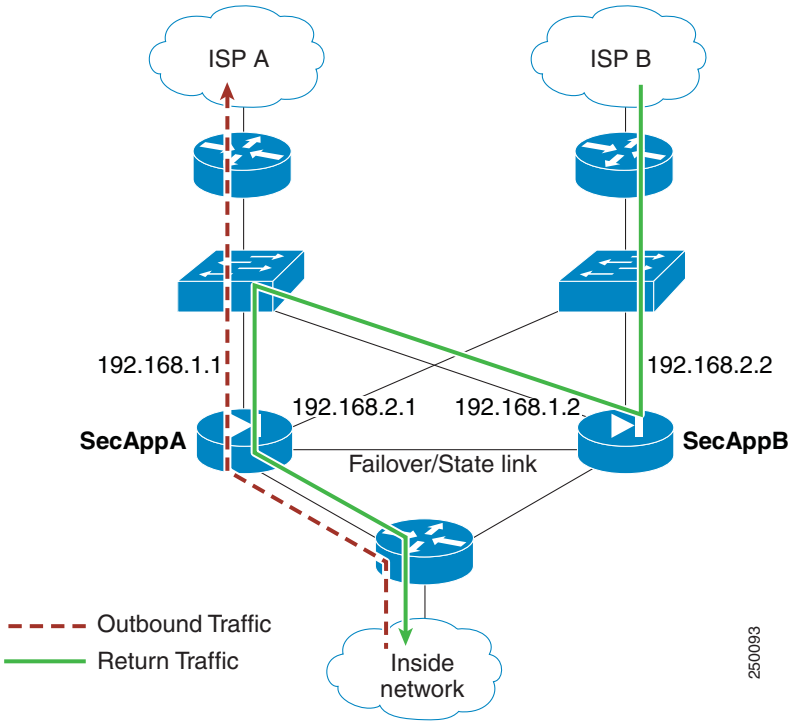
```
hostname/ctx(config)# interface phy_if
hostname/ctx(config-if)# asr-group num
```

Valid values for *num* range from 1 to 32. You need to enter the command for each interface that participates in the asymmetric routing group. You can view the number of ASR packets transmitted, received, or dropped by an interface using the **show interface detail** command. You can have more than one ASR group configured on the security appliance, but only one per interface. Only members of the same ASR group are checked for session information.

Example

Figure 15-1 shows an example of using the **asr-group** command for asymmetric routing support.

Figure 15-1 ASR Example



The two units have the following configuration (configurations show only the relevant commands). The device labeled SecAppA in the diagram is the primary unit in the failover pair.

Example 15-1 Primary Unit System Configuration

```
hostname primary
interface GigabitEthernet0/1
description LAN/STATE Failover Interface
interface GigabitEthernet0/2
no shutdown
interface GigabitEthernet0/3
no shutdown
interface GigabitEthernet0/4
no shutdown
interface GigabitEthernet0/5
no shutdown
failover
failover lan unit primary
failover lan interface folink GigabitEthernet0/1
failover link folink
failover interface ip folink 10.0.4.1 255.255.255.0 standby 10.0.4.11
failover group 1
primary
failover group 2
secondary
admin-context admin
context admin
description admin
allocate-interface GigabitEthernet0/2
allocate-interface GigabitEthernet0/3
config-url flash:/admin.cfg
join-failover-group 1
```

```
context ctx1
description context 1
allocate-interface GigabitEthernet0/4
allocate-interface GigabitEthernet0/5
config-url flash:/ctx1.cfg
join-failover-group 2
```

Example 15-2 admin Context Configuration

```
hostname SecAppA
interface GigabitEthernet0/2
nameif outsideISP-A
security-level 0
ip address 192.168.1.1 255.255.255.0 standby 192.168.1.2
asr-group 1
interface GigabitEthernet0/3
nameif inside
security-level 100
ip address 10.1.0.1 255.255.255.0 standby 10.1.0.11
monitor-interface outside
```

Example 15-3 ctx1 Context Configuration

```
hostname SecAppB
interface GigabitEthernet0/4
nameif outsideISP-B
security-level 0
ip address 192.168.2.2 255.255.255.0 standby 192.168.2.1
asr-group 1
interface GigabitEthernet0/5
nameif inside
security-level 100
ip address 10.2.20.1 255.255.255.0 standby 10.2.20.11
```

Figure 15-1 on page 15-39 shows the ASR support working as follows:

1. An outbound session passes through security appliance SecAppA. It exits interface outsideISP-A (192.168.1.1).
2. Because of asymmetric routing configured somewhere upstream, the return traffic comes back through the interface outsideISP-B (192.168.2.2) on security appliance SecAppB.
3. Normally the return traffic would be dropped because there is no session information for the traffic on interface 192.168.2.2. However, the interface is configure with the command **asr-group 1**. The unit looks for the session on any other interface configured with the same ASR group ID.
4. The session information is found on interface outsideISP-A (192.168.1.2), which is in the standby state on the unit SecAppB. Stateful Failover replicated the session information from SecAppA to SecAppB.
5. Instead of being dropped, the layer 2 header is re-written with information for interface 192.168.1.1 and the traffic is redirected out of the interface 192.168.1.2, where it can then return through the interface on the unit from which it originated (192.168.1.1 on SecAppA). This forwarding continues as needed until the session ends.

Configuring Unit Health Monitoring

The security appliance sends hello packets over the failover interface to monitor unit health. If the standby unit does not receive a hello packet from the active unit for two consecutive polling periods, it sends additional testing packets through the remaining device interfaces. If a hello packet or a response to the interface test packets is not received within the specified hold time, the standby unit becomes active.

You can configure the frequency of hello messages when monitoring unit health. Decreasing the poll time allows a unit failure to be detected more quickly, but consumes more system resources.

To change the unit poll time, enter the following command in global configuration mode:

```
hostname(config)# failover polltime [msec] time [holdtime [msec] time]
```

You can configure the polling frequency from 1 to 15 seconds or, if the optional **msec** keyword is used, from 200 to 999 milliseconds. The hold time determines how long it takes from the time a hello packet is missed to when failover occurs. The hold time must be at least 3 times the poll time. You can configure the hold time from 1 to 45 seconds or, if the optional **msec** keyword is used, from 800 to 990 milliseconds.

Setting the security appliance to use the minimum poll and hold times allows it to detect and respond to unit failures in under a second, but it also increases system resource usage and can cause false failure detection in cases where the networks are congested or where the security appliance is running near full capacity.

Configuring Failover Communication Authentication/Encryption

You can encrypt and authenticate the communication between failover peers by specifying a shared secret or hexadecimal key.

Note On the PIX 500 series security appliance, if you are using the dedicated serial failover cable to connect the units, then communication over the failover link is not encrypted even if a failover key is configured. The failover key only encrypts LAN-based failover communication.

Caution All information sent over the failover and Stateful Failover links is sent in clear text unless you secure the communication with a failover key. If the security appliance is used to terminate VPN tunnels, this information includes any usernames, passwords and preshared keys used for establishing the tunnels. Transmitting this sensitive data in clear text could pose a significant security risk. We recommend securing the failover communication with a failover key if you are using the security appliance to terminate VPN tunnels.

Enter the following command on the active unit of an Active/Standby failover pair or on the unit that has failover group 1 in the active state of an Active/Active failover pair:

```
hostname(config)# failover key {secret | hex key}
```

The *secret* argument specifies a shared secret that is used to generate the encryption key. It can be from 1 to 63 characters. The characters can be any combination of numbers, letters, or punctuation. The **hex** *key* argument specifies a hexadecimal encryption key. The key must be 32 hexadecimal characters (0-9, a-f).

Note To prevent the failover key from being replicated to the peer unit in clear text for an existing failover configuration, disable failover on the active unit (or in the system execution space on the unit that has failover group 1 in the active state), enter the failover key on both units, and then reenable failover. When failover is reenabled, the failover communication is encrypted with the key.

For new LAN-based failover configurations, the **failover key** command should be part of the failover pair bootstrap configuration.

Verifying the Failover Configuration

This section describes how to verify your failover configuration. This section includes the following topics:

- Using the show failover Command, page 15-42
- Viewing Monitored Interfaces, page 15-50
- Displaying the Failover Commands in the Running Configuration, page 15-50
- Testing the Failover Functionality, page 15-51

Using the show failover Command

This section describes the **show failover** command output. On each unit you can verify the failover status by entering the **show failover** command. The information displayed depends upon whether you are using Active/Standby or Active/Active failover.

This section includes the following topics:

- show failover—Active/Standby, page 15-42
- Show Failover—Active/Active, page 15-46

show failover—Active/Standby

The following is sample output from the **show failover** command for Active/Standby Failover. Table 15-10 provides descriptions for the information shown.

```
hostname# show failover

Failover On
Cable status: N/A - LAN-based failover enabled
Failover unit Primary
Failover LAN Interface: fover Ethernet2 (up)
Unit Poll frequency 1 seconds, holdtime 3 seconds
Interface Poll frequency 15 seconds
Interface Policy 1
Monitored Interfaces 2 of 250 maximum
failover replication http
Last Failover at: 22:44:03 UTC Dec 8 2004
        This host: Primary - Active
                Active time: 13434 (sec)
                Interface inside (10.130.9.3): Normal
                Interface outside (10.132.9.3): Normal
        Other host: Secondary - Standby Ready
                Active time: 0 (sec)
                Interface inside (10.130.9.4): Normal
                Interface outside (10.132.9.4): Normal
```

```
Stateful Failover Logical Update Statistics
        Link : fover Ethernet2 (up)
        Stateful Obj    xmit       xerr       rcv        rerr
        General         1950       0          1733       0
        sys cmd         1733       0          1733       0
        up time         0          0          0          0
        RPC services    0          0          0          0
        TCP conn        6          0          0          0
        UDP conn        0          0          0          0
        ARP tbl         106        0          0          0
        Xlate_Timeout   0          0          0          0
        VPN IKE upd     15         0          0          0
        VPN IPSEC upd   90         0          0          0
        VPN CTCP upd    0          0          0          0
        VPN SDI upd     0          0          0          0
        VPN DHCP upd    0          0          0          0
        SIP Session     0          0          0          0

        Logical Update Queue Information
                        Cur     Max     Total
        Recv Q:         0       2       1733
        Xmit Q:         0       2       15225
```

In multiple context mode, using the **show failover** command in a security context displays the failover information for that context. The information is similar to the information shown when using the command in single context mode. Instead of showing the active/standby status of the unit, it displays the active/standby status of the context. Table 15-10 provides descriptions for the information shown.

```
Failover On
Last Failover at: 04:03:11 UTC Jan 4 2003
        This context: Negotiation
                Active time: 1222 (sec)
                Interface outside (192.168.5.121): Normal
                Interface inside (192.168.0.1): Normal
        Peer context: Not Detected
                Active time: 0 (sec)
                Interface outside (192.168.5.131): Normal
                Interface inside (192.168.0.11): Normal

Stateful Failover Logical Update Statistics
        Status: Configured.
        Stateful Obj    xmit       xerr       rcv        rerr
        RPC services    0          0          0          0
        TCP conn        99          0          0          0
        UDP conn        0          0          0          0
        ARP tbl         22          0          0          0
        Xlate_Timeout   0          0          0          0
        GTP PDP         0          0          0          0
        GTP PDPMCB      0          0          0          0
        SIP Session     0          0          0          0
```

Table 15-10 Show Failover Display Description

Field	Options
Failover	• On • Off
Cable status:	• Normal—The cable is connected to both units, and they both have power. • My side not connected—The serial cable is not connected to this unit. It is unknown if the cable is connected to the other unit. • Other side is not connected—The serial cable is connected to this unit, but not to the other unit. • Other side powered off—The other unit is turned off. • N/A—LAN-based failover is enabled.
Failover Unit	Primary or Secondary.
Failover LAN Interface	Displays the logical and physical name of the failover link.
Unit Poll frequency	Displays the number of seconds between hello messages sent to the peer unit and the number of seconds during which the unit must receive a hello message on the failover link before declaring the peer failed.
Interface Poll frequency	*n* seconds The number of seconds you set with the **failover polltime interface** command. The default is 15 seconds.
Interface Policy	Displays the number or percentage of interfaces that must fail to trigger failover.
Monitored Interfaces	Displays the number of interfaces monitored out of the maximum possible.
failover replication http	Displays if HTTP state replication is enabled for Stateful Failover.
Last Failover at:	The date and time of the last failover in the following form: *hh*:*mm*:*ss* **UTC** *DayName Month Day yyyy* UTC (Coordinated Universal Time) is equivalent to GMT (Greenwich Mean Time).
This host: Other host:	For each host, the display shows the following information.
Primary or Secondary	• Active • Standby
Active time:	*n* (sec) The amount of time the unit has been active. This time is cumulative, so the standby unit, if it was active in the past, also shows a value.
slot *x*	Information about the module in the slot or empty.

Table 15-10 Show Failover Display Description (continued)

Field	Options
Interface *name* (*n.n.n.n*):	For each interface, the display shows the IP address currently being used on each unit, as well as one of the following conditions: • Failed—The interface has failed. • No Link—The interface line protocol is down. • Normal—The interface is working correctly. • Link Down—The interface has been administratively shut down. • Unknown—The security appliance cannot determine the status of the interface. • Waiting—Monitoring of the network interface on the other unit has not yet started.
Stateful Failover Logical Update Statistics	The following fields relate to the Stateful Failover feature. If the Link field shows an interface name, the Stateful Failover statistics are shown.
Link	• *interface_name*—The interface used for the Stateful Failover link. • Unconfigured—You are not using Stateful Failover. • up—The interface is up and functioning. • down—The interface is either administratively shutdown or is physically down. • failed—The interface has failed and is not passing stateful data.
Stateful Obj	For each field type, the following statistics are shown. They are counters for the number of state information packets sent between the two units; the fields do not necessarily show active connections through the unit. • xmit—Number of transmitted packets to the other unit. • xerr—Number of errors that occurred while transmitting packets to the other unit. • rcv—Number of received packets. • rerr—Number of errors that occurred while receiving packets from the other unit.
General	Sum of all stateful objects.
sys cmd	Logical update system commands; for example, LOGIN and Stay Alive.
up time	Up time, which the active unit passes to the standby unit.
RPC services	Remote Procedure Call connection information.
TCP conn	TCP connection information.
UDP conn	Dynamic UDP connection information.
ARP tbl	Dynamic ARP table information.
L2BRIDGE tbl	Layer 2 bridge table information (transparent firewall mode only).
Xlate_Timeout	Indicates connection translation timeout information.
VPN IKE upd	IKE connection information.

Table 15-10 Show Failover Display Description (continued)

Field	Options
VPN IPSEC upd	IPSec connection information.
VPN CTCP upd	cTCP tunnel connection information.
VPN SDI upd	SDI AAA connection information.
VPN DHCP upd	Tunneled DHCP connection information.
GTP PDP	GTP PDP update information. This information appears only if inspect GTP is enabled.
GTP PDPMCB	GTP PDPMCB update information. This information appears only if inspect GTP is enabled.
Logical Update Queue Information	For each field type, the following statistics are used: • Cur—Current number of packets • Max—Maximum number of packets • Total—Total number of packets
Recv Q	The status of the receive queue.
Xmit Q	The status of the transmit queue.

Show Failover—Active/Active

The following is sample output from the **show failover** command for Active/Active Failover. Table 15-11 provides descriptions for the information shown.

```
hostname# show failover

Failover On
Failover unit Primary
Failover LAN Interface: third GigabitEthernet0/2 (up)
Unit Poll frequency 1 seconds, holdtime 15 seconds
Interface Poll frequency 4 seconds
Interface Policy 1
Monitored Interfaces 8 of 250 maximum
failover replication http
Group 1 last failover at: 13:40:18 UTC Dec 9 2004
Group 2 last failover at: 13:40:06 UTC Dec 9 2004

  This host:    Primary
  Group 1       State:          Active
                Active time:    2896 (sec)
  Group 2       State:          Standby Ready
                Active time:    0 (sec)

                slot 0: ASA-5530 hw/sw rev (1.0/7.0(0)79) status (Up Sys)
                slot 1: SSM-IDS-20 hw/sw rev (1.0/5.0(0.11)S91(0.11)) status (Up)
                admin Interface outside (10.132.8.5): Normal
                admin Interface third (10.132.9.5): Normal
                admin Interface inside (10.130.8.5): Normal
                admin Interface fourth (10.130.9.5): Normal
                ctx1 Interface outside (10.1.1.1): Normal
                ctx1 Interface inside (10.2.2.1): Normal
                ctx2 Interface outside (10.3.3.2): Normal
                ctx2 Interface inside (10.4.4.2): Normal

  Other host:   Secondary
```

```
  Group 1       State:          Standby Ready
                Active time:    190 (sec)
  Group 2       State:          Active
                Active time:    3322 (sec)

                slot 0: ASA-5530 hw/sw rev (1.0/7.0(0)79) status (Up Sys)
                slot 1: SSM-IDS-20 hw/sw rev (1.0/5.0(0.1)S91(0.1)) status (Up)
                admin Interface outside (10.132.8.6): Normal
                admin Interface third (10.132.9.6): Normal
                admin Interface inside (10.130.8.6): Normal
                admin Interface fourth (10.130.9.6): Normal
                ctx1 Interface outside (10.1.1.2): Normal
                ctx1 Interface inside (10.2.2.2): Normal
                ctx2 Interface outside (10.3.3.1): Normal
                ctx2 Interface inside (10.4.4.1): Normal

Stateful Failover Logical Update Statistics
        Link : third GigabitEthernet0/2 (up)
        Stateful Obj    xmit       xerr       rcv        rerr
        General         1973       0          1895       0
        sys cmd         380        0          380        0
        up time         0          0          0          0
        RPC services    0          0          0          0
        TCP conn        1435       0          1450       0
        UDP conn        0          0          0          0
        ARP tbl         124        0          65         0
        Xlate_Timeout   0          0          0          0
        VPN IKE upd     15         0          0          0
        VPN IPSEC upd   90         0          0          0
        VPN CTCP upd    0          0          0          0
        VPN SDI upd     0          0          0          0
        VPN DHCP upd    0          0          0          0
        SIP Session     0          0          0          0

        Logical Update Queue Information
                        Cur     Max     Total
        Recv Q:         0       1       1895
        Xmit Q:         0       0       1940
```

The following is sample output from the **show failover group** command for Active/Active Failover. The information displayed is similar to that of the **show failover** command, but limited to the specified group. Table 15-11 provides descriptions for the information shown.

```
hostname# show failover group 1

  Last Failover at: 04:09:59 UTC Jan 4 2005

  This host:    Secondary
                State:          Active
                Active time:    186 (sec)

                admin Interface outside (192.168.5.121): Normal
                admin Interface inside (192.168.0.1): Normal


  Other host:   Primary
                State:          Standby
                Active time:    0 (sec)

                admin Interface outside (192.168.5.131): Normal
                admin Interface inside (192.168.0.11): Normal

  Stateful Failover Logical Update Statistics
```

```
Status: Configured.
RPC services    0          0          0          0
TCP conn        33         0          0          0
UDP conn        0          0          0          0
ARP tbl         12         0          0          0
Xlate_Timeout   0          0          0          0
GTP PDP         0          0          0          0
GTP PDPMCB      0          0          0          0
SIP Session    0          0          0          0
```

Table 15-11 Show Failover Display Description

Field	Options
Failover	• On • Off
Failover Unit	Primary or Secondary.
Failover LAN Interface	Displays the logical and physical name of the failover link.
Unit Poll frequency	Displays the number of seconds between hello messages sent to the peer unit and the number of seconds during which the unit must receive a hello message on the failover link before declaring the peer failed.
Interface Poll frequency	*n* seconds The number of seconds you set with the **failover polltime interface** command. The default is 15 seconds.
Interface Policy	Displays the number or percentage of interfaces that must fail before triggering failover.
Monitored Interfaces	Displays the number of interfaces monitored out of the maximum possible.
Group 1 Last Failover at: Group 2 Last Failover at:	The date and time of the last failover for each group in the following form: *hh*:*mm*:*ss* **UTC** *DayName Month Day yyyy* UTC (Coordinated Universal Time) is equivalent to GMT (Greenwich Mean Time).
This host: Other host:	For each host, the display shows the following information.
Role	Primary or Secondary
System State	• Active or Standby Ready • Active Time in seconds
Group 1 State Group 2 State	• Active or Standby Ready • Active Time in seconds
slot *x*	Information about the module in the slot or empty.

Table 15-11 Show Failover Display Description (continued)

Field	Options
context Interface *name* (*n.n.n.n*):	For each interface, the display shows the IP address currently being used on each unit, as well as one of the following conditions: • Failed—The interface has failed. • No link—The interface line protocol is down. • Normal—The interface is working correctly. • Link Down—The interface has been administratively shut down. • Unknown—The security appliance cannot determine the status of the interface. • Waiting—Monitoring of the network interface on the other unit has not yet started.
Stateful Failover Logical Update Statistics	The following fields relate to the Stateful Failover feature. If the Link field shows an interface name, the Stateful Failover statistics are shown.
Link	• *interface_name*—The interface used for the Stateful Failover link. • Unconfigured—You are not using Stateful Failover. • up—The interface is up and functioning. • down—The interface is either administratively shutdown or is physically down. • failed—The interface has failed and is not passing stateful data.
Stateful Obj	For each field type, the following statistics are used. They are counters for the number of state information packets sent between the two units; the fields do not necessarily show active connections through the unit. • xmit—Number of transmitted packets to the other unit • xerr—Number of errors that occurred while transmitting packets to the other unit • rcv—Number of received packets • rerr—Number of errors that occurred while receiving packets from the other unit
General	Sum of all stateful objects.
sys cmd	Logical update system commands; for example, LOGIN and Stay Alive.
up time	Up time, which the active unit passes to the standby unit.
RPC services	Remote Procedure Call connection information.
TCP conn	TCP connection information.
UDP conn	Dynamic UDP connection information.
ARP tbl	Dynamic ARP table information.
L2BRIDGE tbl	Layer 2 bridge table information (transparent firewall mode only).
Xlate_Timeout	Indicates connection translation timeout information.
VPN IKE upd	IKE connection information.

Table 15-11 Show Failover Display Description (continued)

Field	Options
VPN IPSEC upd	IPSec connection information.
VPN CTCP upd	cTCP tunnel connection information.
VPN SDI upd	SDI AAA connection information.
VPN DHCP upd	Tunneled DHCP connection information.
GTP PDP	GTP PDP update information. This information appears only if inspect GTP is enabled.
GTP PDPMCB	GTP PDPMCB update information. This information appears only if inspect GTP is enabled.
Logical Update Queue Information	For each field type, the following statistics are used: • Cur—Current number of packets • Max—Maximum number of packets • Total—Total number of packets
Recv Q	The status of the receive queue.
Xmit Q	The status of the transmit queue.

Viewing Monitored Interfaces

To view the status of monitored interfaces, enter the following command. In single context mode, enter this command in global configuration mode. In multiple context mode, enter this command within a context.

```
primary/context(config)# show monitor-interface
```

For example:

```
hostname/context(config)# show monitor-interface
        This host: Primary - Active
                Interface outside (192.168.1.2): Normal
                Interface inside (10.1.1.91): Normal
        Other host: Secondary - Standby
                Interface outside (192.168.1.3): Normal
                Interface inside (10.1.1.100): Normal
```

Displaying the Failover Commands in the Running Configuration

To view the failover commands in the running configuration, enter the following command:

```
hostname(config)# show running-config failover
```

All of the failover commands are displayed. On units running multiple context mode, enter this command in the system execution space. Entering **show running-config all failover** displays the failover commands in the running configuration and includes commands for which you have not changed the default value.

Testing the Failover Functionality

To test failover functionality, perform the following steps:

Step 1 Test that your active unit or failover group is passing traffic as expected by using FTP (for example) to send a file between hosts on different interfaces.

Step 2 Force a failover to the standby unit by entering the following command:

- For Active/Standby failover, enter the following command on the active unit:

  ```
  hostname(config)# no failover active
  ```

- For Active/Active failover, enter the following command on the unit where the failover group containing the interface connecting your hosts is active:

  ```
  hostname(config)# no failover active group group_id
  ```

Step 3 Use FTP to send another file between the same two hosts.

Step 4 If the test was not successful, enter the **show failover** command to check the failover status.

Step 5 When you are finished, you can restore the unit or failover group to active status by enter the following command:

- For Active/Standby failover, enter the following command on the active unit:

  ```
  hostname(config)# failover active
  ```

- For Active/Active failover, enter the following command on the unit where the failover group containing the interface connecting your hosts is active:

  ```
  hostname(config)# failover active group group_id
  ```

Controlling and Monitoring Failover

This sections describes how to control and monitor failover. This section includes the following topics:

- Forcing Failover, page 15-51
- Disabling Failover, page 15-52
- Restoring a Failed Unit or Failover Group, page 15-52
- Monitoring Failover, page 15-53

Forcing Failover

To force the standby unit or failover group to become active, enter one of the following commands:

- For Active/Standby failover:

  Enter the following command on the standby unit:

  ```
  hostname# failover active
  ```

Or, enter the following command on the active unit:

```
hostname# no failover active
```

- For Active/Active failover:

  Enter the following command in the system execution space of the unit where the failover group is in the standby state:

  ```
  hostname# failover active group group_id
  ```

  Or, enter the following command in the system execution space of the unit where the failover group is in the active state:

  ```
  hostname# no failover active group group_id
  ```

  Entering the following command in the system execution space causes all failover groups to become active:

  ```
  hostname# failover active
  ```

Disabling Failover

To disable failover, enter the following command:

```
hostname(config)# no failover
```

Disabling failover on an Active/Standby pair causes the active and standby state of each unit to be maintained until you restart. For example, the standby unit remains in standby mode so that both units do not start passing traffic. To make the standby unit active (even with failover disabled), see the "Forcing Failover" section on page 15-51.

Disabling failover on an Active/Active failover pair causes the failover groups to remain in the active state on whichever unit they are currently active on, no matter which unit they are configured to prefer. Enter the **no failover** command in the system execution space.

Restoring a Failed Unit or Failover Group

To restore a failed unit to an unfailed state, enter the following command:

```
hostname(config)# failover reset
```

To restore a failed Active/Active failover group to an unfailed state, enter the following command:

```
hostname(config)# failover reset group group_id
```

Restoring a failed unit or group to an unfailed state does not automatically make it active; restored units or groups remain in the standby state until made active by failover (forced or natural). An exception is a failover group configured with the **preempt** command. If previously active, a failover group becomes active if it is configured with the **preempt** command and if the unit on which it failed is the preferred unit.

Monitoring Failover

When a failover occurs, both security appliances send out system messages. This section includes the following topics:

- Failover System Messages, page 15-53
- Debug Messages, page 15-53
- SNMP, page 15-53

Failover System Messages

The security appliance issues a number of system messages related to failover at priority level 2, which indicates a critical condition. To view these messages, see the *Cisco Security Appliance Logging Configuration and System Log Messages* to enable logging and to see descriptions of the system messages.

Note During switchover, failover logically shuts down and then bring up interfaces, generating syslog 411001 and 411002 messages. This is normal activity.

Debug Messages

To see debug messages, enter the **debug fover** command. See the *Cisco Security Appliance Command Reference* for more information.

Note Because debugging output is assigned high priority in the CPU process, it can drastically affect system performance. For this reason, use the **debug fover** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC.

SNMP

To receive SNMP syslog traps for failover, configure the SNMP agent to send SNMP traps to SNMP management stations, define a syslog host, and compile the Cisco syslog MIB into your SNMP management station. See the **snmp-server** and **logging** commands in the *Cisco Security Appliance Command Reference* for more information.

Remote Command Execution

Remote command execution lets you send commands entered at the command line to a specific failover peer.

Because configuration commands are replicated from the active unit or context to the standby unit or context, you can use the **failover exec** command to enter configuration commands on the correct unit, no matter which unit you are logged-in to. For example, if you are logged-in to the standby unit, you can use the **failover exec active** command to send configuration changes to the active unit. Those changes are then replicated to the standby unit. Do not use the **failover exec** command to send configuration commands to the standby unit or context; those configuration changes are not replicated to the active unit and the two configurations will no longer be synchronized.

Output from configuration, exec, and **show** commands is displayed in the current terminal session, so you can use the **failover exec** command to issue **show** commands on a peer unit and view the results in the current terminal.

You must have sufficient privileges to execute a command on the local unit to execute the command on the peer unit.

To send a command to a failover peer, perform the following steps:

Step 1 If you are in multiple context mode, use the **changeto** command to change to the context you want to configure. You cannot change contexts on the failover peer with the **failover exec** command.

If you are in single context mode, skip to the next step.

Step 2 Use the following command to send commands to he specified failover unit:

```
hostname(config)# failover exec {active | mate | standby}
```

Use the **active** or **standby** keyword to cause the command to be executed on the specified unit, even if that unit is the current unit. Use the **mate** keyword to cause the command to be executed on the failover peer.

Commands that cause a command mode change do not change the prompt for the current session. You must use the **show failover exec** command to display the command mode the command is executed in. See Changing Command Modes, page 15-54, for more information.

Changing Command Modes

The **failover exec** command maintains a command mode state that is separate from the command mode of your terminal session. By default, the **failover exec** command mode starts in global configuration mode for the specified device. You can change that command mode by sending the appropriate command (such as the **interface** command) using the **failover exec** command. The session prompt does not change when you change mode using **failover exec**.

For example, if you are logged-in to global configuration mode of the active unit of a failover pair, and you use the **failover exec active** command to change to interface configuration mode, the terminal prompt remains in global configuration mode, but commands entered using **failover exec** are entered in interface configuration mode.

The following examples shows the difference between the terminal session mode and the **failover exec** command mode. In the example, the administrator changes the **failover exec** mode on the active unit to interface configuration mode for the interface GigabitEthernet0/1. After that, all commands entered using **failover exec active** are sent to interface configuration mode for interface GigabitEthernet0/1. The administrator then uses failover exec active to assign an IP address to that interface. Although the prompt indicates global configuration mode, the **failover exec active** mode is in interface configuration mode.

```
hostname(config)# failover exec active interface GigabitEthernet0/1
hostname(config)# failover exec active ip address 192.168.1.1 255.255.255.0 standby
192.168.1.2
hostname(config)# router rip
hostname(config-router)#
```

Changing commands modes for your current session to the device does not affect the command mode used by the **failover exec** command. For example, if you are in interface configuration mode on the active unit, and you have not changed the **failover exec** command mode, the following command would

be executed in global configuration mode. The result would be that your session to the device remains in interface configuration mode, while commands entered using **failover exec active** are sent to router configuration mode for the specified routing process.

```
hostname(config-if)# failover exec active router ospf 100
hostname(config-if)#
```

Use the **show failover exec** command to display the command mode on the specified device in which commands sent with the **failover exec** command are executed. The **show failover exec** command takes the same keywords as the failover exec command: **active**, **mate**, or **standby**. The **failover exec** mode for each device is tracked separately.

For example, the following is sample output from the **show failover exec** command entered on the standby unit:

```
hostname(config)# failover exec active interface GigabitEthernet0/1
hostname(config)# sh failover exec active
Active unit Failover EXEC is at interface sub-command mode

hostname(config)# sh failover exec standby
Standby unit Failover EXEC is at config mode

hostname(config)# sh failover exec mate
Active unit Failover EXEC is at interface sub-command mode
```

Security Considerations

The **failover exec** command uses the failover link to send commands to and receive the output of the command execution from the peer unit. You should use the **failover key** command to encrypt the failover link to prevent eavesdropping or man-in-the-middle attacks.

Limitations of Remote Command Execution

- If you upgrade one unit using the zero-downtime upgrade procedure and not the other, both units must be running software that supports the **failover exec** command for the command to work.
- Command completion and context help is not available for the commands in the *cmd_string* argument.
- In multiple context mode, you can only send commands to the peer context on the peer unit. To send commands to a different context, you must first change to that context on the unit you are logged-in to.
- You cannot use the following commands with the **failover exec** command:
  - **changeto**
  - **debug** (**undebug**)
- If the standby unit is in the failed state, it can still receive commands from the **failover exe**c command if the failure is due to a service card failure; otherwise, the remote command execution will fail.
- You cannot use the **failover exec** command to switch from privileged EXEC mode to global configuration mode on the failover peer. For example, if the current unit is in privileged EXEC mode, and you enter **failover exec mate configure terminal**, the **show failover exec mate** output

will show that the failover exec session is in global configuration mode. However, entering configuration commands for the peer unit using **failover exec** will fail until you enter global configuration mode on the current unit.

- You cannot enter recursive failover exec commands, such as **failover exec mate failover exec mate** *command.*
- Commands that require user input or confirmation must use the **/nonconfirm** option.

Auto Update Server Support in Failover Configurations

You can use Auto Update Server to deploy software images and configuration files to security appliances in an Active/Standby failover configuration. To enable Auto Update on an Active/Standby failover configuration, enter the Auto Update Server configuration on the primary unit in the failover pair. See Configuring Auto Update Support, page 43-19, for more information.

The following restrictions and behaviors apply to Auto Update Server support in failover configurations:

- Only single mode, Active/Standby configurations are supported.
- When loading a new platform software image, the failover pair stops passing traffic.
- When using LAN-based failover, new configurations must not change the failover link configuration. If they do, communication between the units will fail.
- Only the primary unit will perform the call home to the Auto Update Server. The primary unit must be in the active state to call home. If it is not, the security appliance automatically fails over to the primary unit.
- Only the primary unit downloads the software image or configuration file. The software image or configuration is then copied to the secondary unit.
- The interface MAC address and hardware-serial ID is from the primary unit.
- The configuration file stored on the Auto Update Server or HTTP server is for the primary unit only.

Auto Update Process Overview

The following is an overview of the Auto Update process in failover configurations. This process assumes that failover is enabled and operational. The Auto Update process cannot occur if the units are synchronizing configurations, if the standby unit is in the failed state for any reason other than SSM card failure, or if the failover link is down.

1. Both units exchange the platform and ASDM software checksum and version information.
2. The primary unit contacts the Auto Update Server. If the primary unit is not in the active state, the security appliance first fails over to the primary unit and then contacts the Auto Update Server.
3. The Auto Update Server replies with software checksum and URL information.
4. If the primary unit determines that the platform image file needs to be updated for either the active or standby unit, the following occurs:
   a. The primary unit retrieves the appropriate files from the HTTP server using the URL from the Auto Update Server.
   b. The primary unit copies the image to the standby unit and then updates the image on itself.
   c. If both units have new image, the secondary (standby) unit is reloaded first.

- If hitless upgrade can be performed when secondary unit boots, then the secondary unit becomes the active unit and the primary unit reloads. The primary unit becomes the active unit when it has finished loading.
- If hitless upgrade cannot be performed when the standby unit boots, then both units reload at the same time.

d. If only the secondary (standby) unit has new image, then only the secondary unit reloads. The primary unit waits until the secondary unit finishes reloading.

e. If only the primary (active) unit has new image, the secondary unit becomes the active unit, and the primary unit reloads.

f. The update process starts again at step 1.

5. If the security appliance determines that the ASDM file needs to be updated for either the primary or secondary unit, the following occurs:
   a. The primary unit retrieves the ASDM image file from the HTTP server using the URL provided by the Auto Update Server.
   b. The primary unit copies the ASDM image to the standby unit, if needed.
   c. The primary unit updates the ASDM image on itself.
   d. The update process starts again at step 1.
6. If the primary unit determines that the configuration needs to be updated, the following occurs:
   a. The primary unit retrieves the configuration file from the using the specified URL.
   b. The new configuration replaces the old configuration on both units simultaneously.
   c. The update process begins again at step 1.
7. If the checksums match for all image and configuration files, no updates are required. The process ends until the next poll time.

Monitoring the Auto Update Process

You can use the **debug auto-update client** or **debug fover cmd-exe** commands to display the actions performed during the Auto Update process. The following is sample output from the **debug auto-update client** command.

```
Auto-update client: Sent DeviceDetails to /cgi-bin/dda.pl of server 192.168.0.21
Auto-update client: Processing UpdateInfo from server 192.168.0.21
   Component: asdm, URL: http://192.168.0.21/asdm.bint, checksum:
0x94bced0261cc992ae710faf8d244cf32
   Component: config, URL: http://192.168.0.21/config-rms.xml, checksum:
0x67358553572688a805a155af312f6898
   Component: image, URL: http://192.168.0.21/cdisk73.bin, checksum:
0x6d091b43ce96243e29a62f2330139419
Auto-update client: need to update img, act: yes, stby yes
name
ciscoasa(config)# Auto-update client: update img on stby unit...
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 1, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 1001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 1501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 2001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 2501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 3001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 3501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 4001, len = 1024
```

```
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 4501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 5001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 5501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 6001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 6501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 7001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 7501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 8001, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 8501, len = 1024
auto-update: Fover copyfile, seq = 4 type = 1, pseq = 9001, len = 1024
auto-update: Fover file copy waiting at clock tick 6129280
fover_parse: Rcvd file copy ack, ret = 0, seq = 4
auto-update: Fover filecopy returns value: 0 at clock tick 6150260, upd time 145980 msecs
Auto-update client: update img on active unit...
fover_parse: Rcvd image info from mate
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
auto-update: HA safe reload: reload active waiting with mate state: 20
Beginning configuration replication: Sending to mate.
auto-update: HA safe reload: reload active waiting with mate state: 50
auto-update: HA safe reload: reload active waiting with mate state: 50

auto-update: HA safe reload: reload active waiting with mate state: 80
        Sauto-update: HA safe reload: reload active unit at clock tick: 6266860
Auto-update client: Succeeded: Image, version: 0x6d091b43ce96243e29a62f2330139419
```

The following system log message is generated if the Auto Update process fails:

```
%PIX|ASA4-612002: Auto Update failed: file version: version reason: reason
```

The *file* is “image”, “asdm”, or “configuration”, depending on which update failed. The *version* is the version number of the update. And the *reason* is the reason the update failed.



CHAPTER 16

Using Modular Policy Framework

This chapter describes how to use Modular Policy Framework to create security policies for TCP and general connection settings, inspections, IPS, CSC, and QoS. This chapter includes the following sections:

- Information About Modular Policy Framework, page 16-1
- Identifying Traffic (Layer 3/4 Class Map), page 16-4
- Configuring Special Actions for Application Inspections (Inspection Policy Map), page 16-8
- Defining Actions (Layer 3/4 Policy Map), page 16-16
- Applying Actions to an Interface (Service Policy), page 16-23
- Modular Policy Framework Examples, page 16-24

Information About Modular Policy Framework

Modular Policy Framework provides a consistent and flexible way to configure security appliance features. For example, you can use Modular Policy Framework to create a timeout configuration that is specific to a particular TCP application, as opposed to one that applies to all TCP applications. This section includes the following topics:

- Modular Policy Framework Supported Features, page 16-1
- Modular Policy Framework Configuration Overview, page 16-2
- Default Global Policy, page 16-3

Modular Policy Framework Supported Features

Modular Policy Framework supports the following features:

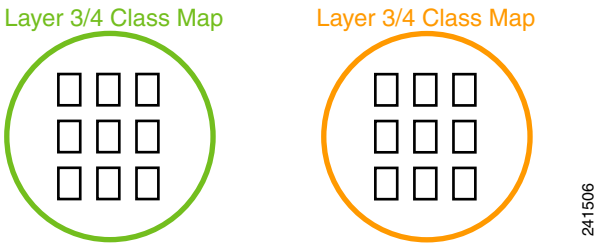
- QoS input policing—See Chapter 25, "Configuring QoS."
- TCP normalization, TCP and UDP connection limits and timeouts, and TCP sequence number randomization—See the "Configuring TCP Normalization" section on page 24-12, and the "Configuring Connection Limits and Timeouts" section on page 24-17.
- CSC—See the "Managing the CSC SSM" section on page 23-9.
- Application inspection (multiple types)—See Chapter 26, "Configuring Application Layer Protocol Inspection."
- IPS—See the "Managing the AIP SSM" section on page 23-1.

- QoS output policing—See Chapter 25, “Configuring QoS.”
- QoS standard priority queue—See Chapter 25, “Configuring QoS.”
- QoS traffic shaping, hierarchical priority queue—See Chapter 25, “Configuring QoS.”

Modular Policy Framework Configuration Overview

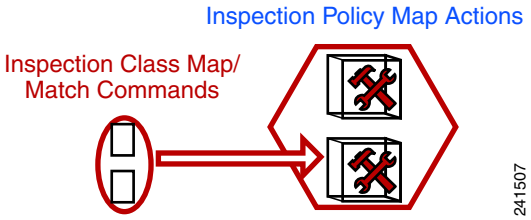
Configuring Modular Policy Framework consists of the following tasks:

1. Identify the traffic on which you want to perform Modular Policy Framework actions by creating Layer 3/4 class maps. For example, you might want to perform actions on all traffic that passes through the security appliance; or you might only want to perform certain actions on traffic from 10.1.1.0/24 to any destination address.



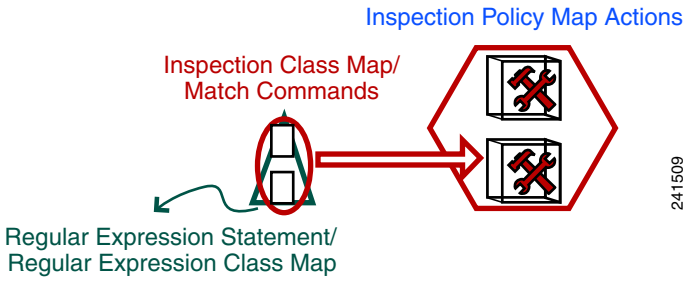
See the “Identifying Traffic (Layer 3/4 Class Map)” section on page 16-4.

2. If one of the actions you want to perform is application inspection, and you want to perform additional actions on some inspection traffic, then create an inspection policy map. The inspection policy map identifies the traffic and specifies what to do with it. For example, you might want to drop all HTTP requests with a body length greater than 1000 bytes.



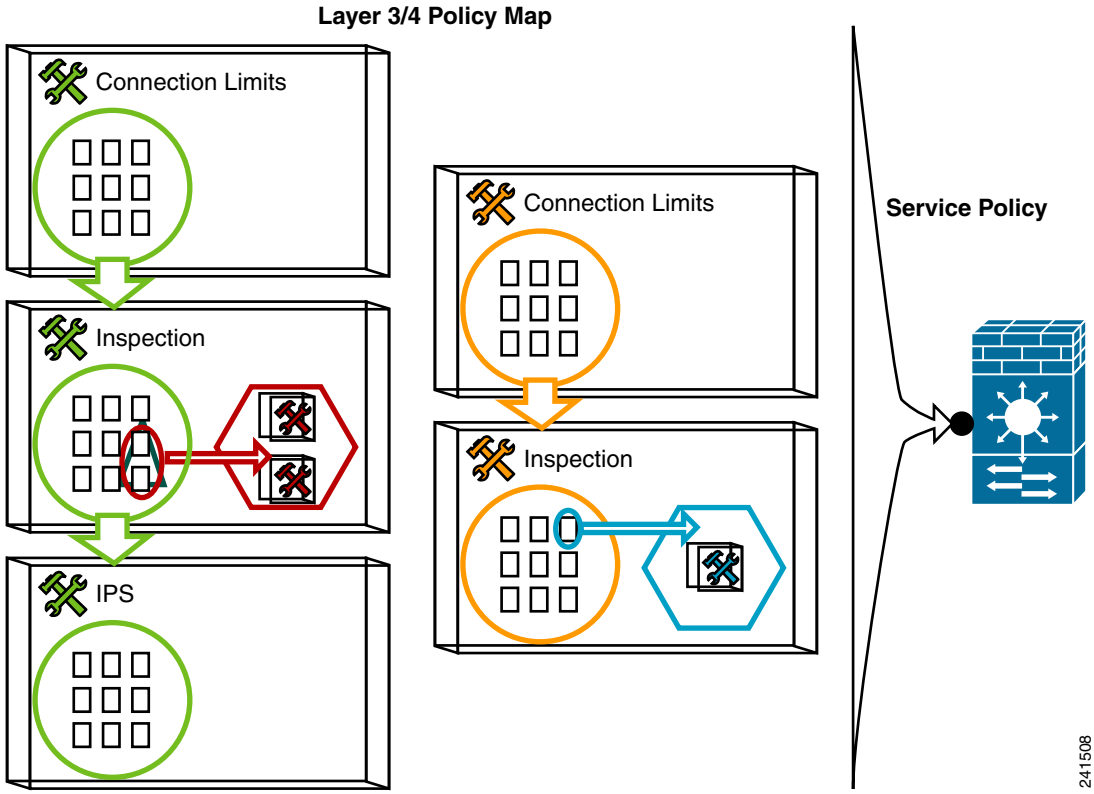
You can create a self-contained inspection policy map that identifies the traffic directly with **match** commands, or you can create an inspection class map for reuse or for more complicated matching. See the “Defining Actions in an Inspection Policy Map” section on page 16-9 and the “Identifying Traffic in an Inspection Class Map” section on page 16-12.

3. If you want to match text with a regular expression within inspected packets, you can create a regular expression or a group of regular expressions (a regular expression class map). Then, when you define the traffic to match for the inspection policy map, you can call on an existing regular expression. For example, you might want to drop all HTTP requests with a URL including the text “example.com.”



See the "Creating a Regular Expression" section on page 16-13 and the "Creating a Regular Expression Class Map" section on page 16-16.

4. Define the actions you want to perform on each Layer 3/4 class map by creating a Layer 3/4 policy map. Then, determine on which interfaces you want to apply the policy map using a service policy.



See the "Defining Actions (Layer 3/4 Policy Map)" section on page 16-16 and the "Applying Actions to an Interface (Service Policy)" section on page 16-23.

Default Global Policy

By default, the configuration includes a policy that matches all default application inspection traffic and applies certain inspections to the traffic on all interfaces (a global policy). Not all inspections are enabled by default. You can only apply one global policy, so if you want to alter the global policy, you need to either edit the default policy or disable it and apply a new one. (An interface policy overrides the global policy for a particular feature.)

The default policy configuration includes the following commands:

```
class-map inspection_default
 match default-inspection-traffic
policy-map type inspect dns preset_dns_map
 parameters
  message-length maximum 512
policy-map global_policy
 class inspection_default
  inspect dns preset_dns_map
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect rsh
  inspect rtsp
  inspect esmtp
  inspect sqlnet
  inspect skinny
  inspect sunrpc
  inspect xdmcp
  inspect sip
  inspect netbios
  inspect tftp
service-policy global_policy global
```

Note See the "Incompatibility of Certain Feature Actions" section on page 16-20 for more information about the special **match default-inspection-traffic** command used in the default class map.

Identifying Traffic (Layer 3/4 Class Map)

A Layer 3/4 class map identifies Layer 3 and 4 traffic to which you want to apply actions. You can create multiple Layer 3/4 class maps for each Layer 3/4 policy map.

This section includes the following topics:

- Default Class Maps, page 16-4
- Maximum Class Maps, page 16-5
- Creating a Layer 3/4 Class Map for Through Traffic, page 16-5
- Creating a Layer 3/4 Class Map for Management Traffic, page 16-7

Default Class Maps

The configuration includes a default Layer 3/4 class map that the security appliance uses in the default global policy. It is called **inspection_default** and matches the default inspection traffic:

```
class-map inspection_default
 match default-inspection-traffic
```

Note See the "Incompatibility of Certain Feature Actions" section on page 16-20 for more information about the special **match default-inspection-traffic** command used in the default class map.

Another class map that exists in the default configuration is called class-default, and it matches all traffic:

```
class-map class-default
 match any
```

This class map appears at the end of all Layer 3/4 policy maps and essentially tells the security appliance to not perform any actions on all other traffic. You can use the class-default class map if desired, rather than making your own **match any** class map. In fact, some features are only available for class-default, such as QoS traffic shaping.

Maximum Class Maps

The maximum number of class maps of all types is 255 in single mode or per context in multiple mode. Class maps include the following types:

- Layer 3/4 class maps (for through traffic and management traffic)
- Inspection class maps
- Regular expression class maps
- **match** commands used directly underneath an inspection policy map

This limit also includes default class maps of all types. See the "Default Class Maps" section on page 16-4.

Creating a Layer 3/4 Class Map for Through Traffic

A Layer 3/4 class map matches traffic based on protocols, ports, IP addresses and other Layer 3 or 4 attributes.

To define a Layer 3/4 class map, perform the following steps:

Step 1 Create a Layer 3/4 class map by entering the following command:

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

Where *class_map_name* is a string up to 40 characters in length. The name "class-default" is reserved. All types of class maps use the same name space, so you cannot reuse a name already used by another type of class map. The CLI enters class-map configuration mode.

Step 2 (Optional) Add a description to the class map by entering the following command:

```
hostname(config-cmap)# description string
```

Step 3 Define the traffic to include in the class by matching one of the following characteristics. Unless otherwise specified, you can include only one **match** command in the class map.

- Any traffic—The class map matches all traffic.

  ```
  hostname(config-cmap)# match any
  ```

- Access list—The class map matches traffic specified by an extended access list. If the security appliance is operating in transparent firewall mode, you can use an EtherType access list.

  ```
  hostname(config-cmap)# match access-list access_list_name
  ```

For more information about creating access lists, see the "Adding an Extended Access List" section on page 18-5 or the "Adding an EtherType Access List" section on page 18-8.

For information about creating access lists with NAT, see the "IP Addresses Used for Access Lists When You Use NAT" section on page 18-3.

- TCP or UDP destination ports—The class map matches a single port or a contiguous range of ports.

```
hostname(config-cmap)# match port {tcp | udp} {eq port_num | range port_num port_num}
```

Tip For applications that use multiple, non-contiguous ports, use the **match access-list** command and define an ACE to match each port.

For a list of ports you can specify, see the "TCP and UDP Ports" section on page C-11.

For example, enter the following command to match TCP packets on port 80 (HTTP):

```
hostname(config-cmap)# match tcp eq 80
```

- Default traffic for inspection—The class map matches the default TCP and UDP ports used by all applications that the security appliance can inspect.

```
hostname(config-cmap)# match default-inspection-traffic
```

This command, which is used in the default global policy, is a special CLI shortcut that when used in a policy map, ensures that the correct inspection is applied to each packet, based on the destination port of the traffic. For example, when UDP traffic for port 69 reaches the security appliance, then the security appliance applies the TFTP inspection; when TCP traffic for port 21 arrives, then the security appliance applies the FTP inspection. So in this case only, you can configure multiple inspections for the same class map (with the exception of WAAS inspection, which can be configured with other inspections. See the "Incompatibility of Certain Feature Actions" section on page 16-20 for more information about combining actions). Normally, the security appliance does not use the port number to determine the inspection applied, thus giving you the flexibility to apply inspections to non-standard ports, for example.

See the "Default Inspection Policy" section on page 26-3 for a list of default ports. Not all applications whose ports are included in the **match default-inspection-traffic** command are enabled by default in the policy map.

You can specify a **match access-list** command along with the **match default-inspection-traffic** command to narrow the matched traffic. Because the **match default-inspection-traffic** command specifies the ports to match, any ports in the access list are ignored.

- DSCP value in an IP header—The class map matches up to eight DSCP values.

```
hostname(config-cmap)# match dscp value1 [value2] [...] [value8]
```

For example, enter the following:

```
hostname(config-cmap)# match dscp af43 cs1 ef
```

- Precedence—The class map matches up to four precedence values, represented by the TOS byte in the IP header.

```
hostname(config-cmap)# match precedence value1 [value2] [value3] [value4]
```

where *value1* through *value4* can be 0 to 7, corresponding to the possible precedences.

- RTP traffic—The class map matches RTP traffic.

```
hostname(config-cmap)# match rtp starting_port range
```

The *starting_port* specifies an even-numbered UDP destination port between 2000 and 65534. The *range* specifies the number of additional UDP ports to match above the *starting_port*, between 0 and 16383.

- Tunnel group traffic—The class map matches traffic for a tunnel group to which you want to apply QoS.

  ```
  hostname(config-cmap)# match tunnel-group name
  ```

  You can also specify one other **match** command to refine the traffic match. You can specify any of the preceding commands, except for the **match any**, **match access-list**, or **match default-inspection-traffic** commands. Or you can enter the following command to police each flow:

  ```
  hostname(config-cmap)# match flow ip destination address
  ```

  All traffic going to a unique IP destination address is considered a flow.

The following is an example for the **class-map** command:

```
hostname(config)# access-list udp permit udp any any
hostname(config)# access-list tcp permit tcp any any
hostname(config)# access-list host_foo permit ip any 10.1.1.1 255.255.255.255

hostname(config)# class-map all_udp
hostname(config-cmap)# description "This class-map matches all UDP traffic"
hostname(config-cmap)# match access-list udp

hostname(config-cmap)# class-map all_tcp
hostname(config-cmap)# description "This class-map matches all TCP traffic"
hostname(config-cmap)# match access-list tcp

hostname(config-cmap)# class-map all_http
hostname(config-cmap)# description "This class-map matches all HTTP traffic"
hostname(config-cmap)# match port tcp eq http

hostname(config-cmap)# class-map to_server
hostname(config-cmap)# description "This class-map matches all traffic to server 10.1.1.1"
hostname(config-cmap)# match access-list host_foo
```

Creating a Layer 3/4 Class Map for Management Traffic

For management traffic to the security appliance, you might want to perform actions specific to this kind of traffic. You can specify a management class map that can match an access list or TCP or UDP ports. The types of actions available for a management class map in the policy map are specialized for management traffic. Namely, this type of class map lets you inspect RADIUS accounting traffic and set connection limits.

To create a class map for management traffic to the security appliance, perform the following steps:

Step 1 Create a class map by entering the following command:

```
hostname(config)# class-map type management class_map_name
hostname(config-cmap)#
```

Where *class_map_name* is a string up to 40 characters in length. The name "class-default" is reserved. All types of class maps use the same name space, so you cannot reuse a name already used by another type of class map. The CLI enters class-map configuration mode.

Step 2 (Optional) Add a description to the class map by entering the following command:

```
hostname(config-cmap)# description string
```

Step 3 Define the traffic to include in the class by matching one of the following characteristics. You can include only one **match** command in the class map.

- Access list—The class map matches traffic specified by an extended access list. If the security appliance is operating in transparent firewall mode, you can use an EtherType access list.

  ```
  hostname(config-cmap)# match access-list access_list_name
  ```

  For more information about creating access lists, see the "Adding an Extended Access List" section on page 18-5 or the "Adding an EtherType Access List" section on page 18-8.

  For information about creating access lists with NAT, see the "IP Addresses Used for Access Lists When You Use NAT" section on page 18-3.

- TCP or UDP destination ports—The class map matches a single port or a contiguous range of ports.

  ```
  hostname(config-cmap)# match port {tcp | udp} {eq port_num | range port_num port_num}
  ```

Tip For applications that use multiple, non-contiguous ports, use the **match access-list** command and define an ACE to match each port.

For a list of ports you can specify, see the "TCP and UDP Ports" section on page C-11.

For example, enter the following command to match TCP packets on port 80 (HTTP):

```
hostname(config-cmap)# match tcp eq 80
```

Configuring Special Actions for Application Inspections (Inspection Policy Map)

Modular Policy Framework lets you configure special actions for many application inspections. When you enable an inspection engine in the Layer 3/4 policy map, you can also optionally enable actions as defined in an *inspection policy map*. When the inspection policy map matches traffic within the Layer 3/4 class map for which you have defined an inspection action, then that subset of traffic will be acted upon as specified (for example, dropped or rate-limited).

This section includes the following topics:

- Inspection Policy Map Overview, page 16-9
- Defining Actions in an Inspection Policy Map, page 16-9
- Identifying Traffic in an Inspection Class Map, page 16-12
- Creating a Regular Expression, page 16-13
- Creating a Regular Expression Class Map, page 16-16

Inspection Policy Map Overview

See the "Configuring Application Inspection" section on page 26-5 for a list of applications that support inspection policy maps.

An inspection policy map consists of one or more of the following elements. The exact options available for an inspection policy map depends on the application.

- Traffic matching command—You can define a traffic matching command directly in the inspection policy map to match application traffic to criteria specific to the application, such as a URL string, for which you then enable actions.
  - Some traffic matching commands can specify regular expressions to match text inside a packet. Be sure to create and test the regular expressions before you configure the policy map, either singly or grouped together in a regular expression class map.
- Inspection class map—(Not available for all applications. See the CLI help for a list of supported applications.) An inspection class map includes traffic matching commands that match application traffic with criteria specific to the application, such as a URL string. You then identify the class map in the policy map and enable actions. The difference between creating a class map and defining the traffic match directly in the inspection policy map is that you can create more complex match criteria and you can reuse class maps.
  - Some traffic matching commands can specify regular expressions to match text inside a packet. Be sure to create and test the regular expressions before you configure the policy map, either singly or grouped together in a regular expression class map.
- Parameters—Parameters affect the behavior of the inspection engine.

The default inspection policy map configuration includes the following commands, which sets the maximum message length for DNS packets to be 512 bytes:

```
policy-map type inspect dns preset_dns_map
 parameters
  message-length maximum 512
```

Note There are other default inspection policy maps such as **policy-map type inspect esmtp _default_esmtp_map**. These default policy maps are created implicitly by the command **inspect** *protocol*. For example, **inspect esmtp** implicitly uses the policy map "_default_esmtp_map." All the default policy maps can be shown by using the **show running-config all policy-map** command.

Defining Actions in an Inspection Policy Map

When you enable an inspection engine in the Layer 3/4 policy map, you can also optionally enable actions as defined in an inspection policy map.

To create an inspection policy map, perform the following steps:

Step 1 (Optional) Create an inspection class map according to the "Identifying Traffic in an Inspection Class Map" section on page 16-12. Alternatively, you can identify the traffic directly within the policy map.

Step 2 To create the inspection policy map, enter the following command:

```
hostname(config)# policy-map type inspect application policy_map_name
hostname(config-pmap)#
```

See the "Configuring Application Inspection" section on page 26-5 for a list of applications that support inspection policy maps.

The *policy_map_name* argument is the name of the policy map up to 40 characters in length. All types of policy maps use the same name space, so you cannot reuse a name already used by another type of policy map. The CLI enters policy-map configuration mode.

Step 3 To apply actions to matching traffic, perform the following steps:

a. Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the inspection class map that you created in the "Identifying Traffic in an Inspection Class Map" section on page 16-12 by entering the following command:

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

  Not all applications support inspection class maps.

- Specify traffic directly in the policy map using one of the **match** commands described for each application in Chapter 26, "Configuring Application Layer Protocol Inspection." If you use a **match not** command, then any traffic that matches the criterion in the **match not** command does not have the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
hostname(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

Not all options are available for each application. Other actions specific to the application might also be available. See Chapter 26, "Configuring Application Layer Protocol Inspection," for the exact options available.

The **drop** keyword drops all packets that match.

The **send-protocol-error** keyword sends a protocol error message.

The **drop-connection** keyword drops the packet and closes the connection.

The **mask** keyword masks out the matching portion of the packet.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

The **rate-limit** *message_rate* argument limits the rate of messages.

Note You can specify multiple **class** or **match** commands in the policy map.

If a packet matches multiple different **match** or **class** commands, then the order in which the security appliance applies the actions is determined by internal security appliance rules, and not by the order they are added to the policy map. The internal rules are determined by the application type and the logical progression of parsing a packet, and are not user-configurable. For example for HTTP traffic, parsing a Request Method field precedes parsing the Header Host Length field; an action for the Request Method field occurs before the action for the Header Host Length field. For example, the following match commands can be entered in any order, but the **match request method get** command is matched first.

```
match request header host length gt 100
  reset
match request method get
  log
```

If an action drops a packet, then no further actions are performed in the inspection policy map. For example, if the first action is to reset the connection, then it will never match any further **match** or **class** commands. If the first action is to log the packet, then a second action, such as resetting the connection, can occur. (You can configure both the **reset** (or **drop-connection**, and so on.) and the **log** action for the same **match** or **class** command, in which case the packet is logged before it is reset for a given match.)

If a packet matches multiple **match** or **class** commands that are the same, then they are matched in the order they appear in the policy map. For example, for a packet with the header length of 1001, it will match the first command below, and be logged, and then will match the second command and be reset. If you reverse the order of the two **match** commands, then the packet will be dropped and the connection reset before it can match the second **match** command; it will never be logged.

```
match request header length gt 100
  log
match request header length gt 1000
  reset
```

A class map is determined to be the same type as another class map or **match** command based on the lowest priority **match** command in the class map (the priority is based on the internal rules). If a class map has the same type of lowest priority **match** command as another class map, then the class maps are matched according to the order they are added to the policy map. If the lowest priority command for each class map is different, then the class map with the higher priority **match** command is matched first. For example, the following three class maps contain two types of **match** commands: **match request-cmd** (higher priority) and **match filename** (lower priority). The ftp3 class map includes both commands, but it is ranked according to the lowest priority command, **match filename**. The ftp1 class map includes the highest priority command, so it is matched first, regardless of the order in the policy map. The ftp3 class map is ranked as being of the same priority as the ftp2 class map, which also contains the **match filename** command. They are matched according to the order in the policy map: ftp3 and then ftp2.

```
class-map type inspect ftp match-all ftp1
  match request-cmd get
class-map type inspect ftp match-all ftp2
  match filename regex abc
class-map type inspect ftp match-all ftp3
  match request-cmd get
  match filename regex abc

policy-map type inspect ftp ftp
  class ftp3
    log
  class ftp2
    log
  class ftp1
    log
```

Step 4 To configure parameters that affect the inspection engine, enter the following command:

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

The CLI enters parameters configuration mode. For the parameters available for each application, see Chapter 26, “Configuring Application Layer Protocol Inspection.”

The following is an example of an HTTP inspection policy map and the related class maps. This policy map is activated by the Layer 3/4 policy map, which is enabled by the service policy.

```
hostname(config)# regex url_example example\.com
```

```
hostname(config)# regex url_example2 example2\.com
hostname(config)# class-map type regex match-any URLs
hostname(config-cmap)# match regex url_example
hostname(config-cmap)# match regex url_example2

hostname(config-cmap)# class-map type inspect http match-all http-traffic
hostname(config-cmap)# match req-resp content-type mismatch
hostname(config-cmap)# match request body length gt 1000
hostname(config-cmap)# match not request uri regex class URLs

hostname(config-cmap)# policy-map type inspect http http-map1
hostname(config-pmap)# class http-traffic
hostname(config-pmap-c)# drop-connection log
hostname(config-pmap-c)# match req-resp content-type mismatch
hostname(config-pmap-c)# reset log
hostname(config-pmap-c)# parameters
hostname(config-pmap-p)# protocol-violation action log

hostname(config-pmap-p)# policy-map test
hostname(config-pmap)# class test (a Layer 3/4 class map not shown)
hostname(config-pmap-c)# inspect http http-map1

hostname(config-pmap-c)# service-policy test interface outside
```

Identifying Traffic in an Inspection Class Map

This type of class map allows you to match criteria that is specific to an application. For example, for DNS traffic, you can match the domain name in a DNS query.

Note Not all applications support inspection class maps. See the CLI help for a list of supported applications.

A class map groups multiple traffic matches (in a match-all class map), or lets you match any of a list of matches (in a match-any class map). The difference between creating a class map and defining the traffic match directly in the inspection policy map is that the class map lets you group multiple match commands, and you can reuse class maps. For the traffic that you identify in this class map, you can specify actions such as dropping, resetting, and/or logging the connection in the inspection policy map. If you want to perform different actions on different types of traffic, you should identify the traffic directly in the policy map.

To define an inspection class map, perform the following steps:

Step 1 (Optional) If you want to match based on a regular expression, see the "Creating a Regular Expression" section on page 16-13 and the "Creating a Regular Expression Class Map" section on page 16-16.

Step 2 Create a class map by entering the following command:

```
hostname(config)# class-map type inspect application [match-all | match-any]
class_map_name
hostname(config-cmap)#
```

Where the *application* is the application you want to inspect. For supported applications, see the CLI help for a list of supported applications or see Chapter 26, "Configuring Application Layer Protocol Inspection."

The *class_map_name* argument is the name of the class map up to 40 characters in length.

The **match-all** keyword is the default, and specifies that traffic must match all criteria to match the class map.

The **match-any** keyword specifies that the traffic matches the class map if it matches at least one of the criteria.

The CLI enters class-map configuration mode, where you can enter one or more **match** commands.

Step 3 (Optional) To add a description to the class map, enter the following command:

```
hostname(config-cmap)# description string
```

Step 4 Define the traffic to include in the class by entering one or more **match** commands available for your application.

To specify traffic that should not match the class map, use the **match not** command. For example, if the **match not** command specifies the string "example.com," then any traffic that includes "example.com" does not match the class map.

To see the **match** commands available for each application, see Chapter 26, "Configuring Application Layer Protocol Inspection."

The following example creates an HTTP class map that must match all criteria:

```
hostname(config-cmap)# class-map type inspect http match-all http-traffic
hostname(config-cmap)# match req-resp content-type mismatch
hostname(config-cmap)# match request body length gt 1000
hostname(config-cmap)# match not request uri regex class URLs
```

The following example creates an HTTP class map that can match any of the criteria:

```
hostname(config-cmap)# class-map type inspect http match-any monitor-http
hostname(config-cmap)# match request method get
hostname(config-cmap)# match request method put
hostname(config-cmap)# match request method post
```

Creating a Regular Expression

A regular expression matches text strings either literally as an exact string, or by using *metacharacters* so you can match multiple variants of a text string. You can use a regular expression to match the content of certain application traffic; for example, you can match a URL string inside an HTTP packet.

Use **Ctrl+V** to escape all of the special characters in the CLI, such as question mark (?) or a tab. For example, type **d[Ctrl+V]?g** to enter **d?g** in the configuration.

See the **regex** command in the *Cisco Security Appliance Command Reference* for performance impact information when matching a regular expression to packets.

Note As an optimization, the security appliance searches on the deobfuscated URL. Deobfuscation compresses multiple forward slashes (/) into a single slash. For strings that commonly use double slashes, like "http://", be sure to search for "http:/" instead.

Table 16-1 lists the metacharacters that have special meanings.

Table 16-1 regex Metacharacters

Character	Description	Notes
.	Dot	Matches any single character. For example, **d.g** matches dog, dag, dtg, and any word that contains those characters, such as doggonnit.
(*exp*)	Subexpression	A subexpression segregates characters from surrounding characters, so that you can use other metacharacters on the subexpression. For example, **d(o|a)g** matches dog and dag, but **do|ag** matches do and ag. A subexpression can also be used with repeat quantifiers to differentiate the characters meant for repetition. For example, **ab(xy){3}z** matches abxyxyxyz.
|	Alternation	Matches either expression it separates. For example, **dog|cat** matches dog or cat.
?	Question mark	A quantifier that indicates that there are 0 or 1 of the previous expression. For example, **lo?se** matches lse or lose. **Note** You must enter **Ctrl+V** and then the question mark or else the help function is invoked.
*	Asterisk	A quantifier that indicates that there are 0, 1 or any number of the previous expression. For example, **lo*se** matches lse, lose, loose, and so on.
+	Plus	A quantifier that indicates that there is at least 1 of the previous expression. For example, **lo+se** matches lose and loose, but not lse.
{*x*} or {*x*,}	Minimum repeat quantifier	Repeat at least *x* times. For example, **ab(xy){2,}z** matches abxyxyz, abxyxyxyz, and so on.
[*abc*]	Character class	Matches any character in the brackets. For example, **[abc]** matches a, b, or c.
[^*abc*]	Negated character class	Matches a single character that is not contained within the brackets. For example, **[^abc]** matches any character other than a, b, or c. **[^A-Z]** matches any single character that is not an uppercase letter.
[*a*-*c*]	Character range class	Matches any character in the range. **[a-z]** matches any lowercase letter. You can mix characters and ranges: **[abcq-z]** matches a, b, c, q, r, s, t, u, v, w, x, y, z, and so does [**a-cq-z**]. The dash (-) character is literal only if it is the last or the first character within the brackets: **[abc-]** or **[-abc]**.
"""	Quotation marks	Preserves trailing or leading spaces in the string. For example, **" test"** preserves the leading space when it looks for a match.
^	Caret	Specifies the beginning of a line.
\	Escape character	When used with a metacharacter, matches a literal character. For example, **\[** matches the left square bracket.

Table 16-1 regex Metacharacters (continued)

Character	Description	Notes
char	Character	When character is not a metacharacter, matches the literal character.
\r	Carriage return	Matches a carriage return 0x0d.
\n	Newline	Matches a new line 0x0a.
\t	Tab	Matches a tab 0x09.
\f	Formfeed	Matches a form feed 0x0c.
\x*NN*	Escaped hexadecimal number	Matches an ASCII character using hexadecimal (exactly two digits).
NNN	Escaped octal number	Matches an ASCII character as octal (exactly three digits). For example, the character 040 represents a space.

To test and create a regular expression, perform the following steps:

Step 1 To test a regular expression to make sure it matches what you think it will match, enter the following command:

```
hostname(config)# test regex input_text regular_expression
```

Where the *input_text* argument is a string you want to match using the regular expression, up to 201 characters in length.

The *regular_expression* argument can be up to 100 characters in length.

Use **Ctrl+V** to escape all of the special characters in the CLI. For example, to enter a tab in the input text in the **test regex** command, you must enter **test regex "test[Ctrl+V Tab]" "test\t"**.

If the regular expression matches the input text, you see the following message:

```
INFO: Regular expression match succeeded.
```

If the regular expression does not match the input text, you see the following message:

```
INFO: Regular expression match failed.
```

Step 2 To add a regular expression after you tested it, enter the following command:

```
hostname(config)# regex name regular_expression
```

Where the *name* argument can be up to 40 characters in length.

The *regular_expression* argument can be up to 100 characters in length.

The following example creates two regular expressions for use in an inspection policy map:

```
hostname(config)# regex url_example example\.com
hostname(config)# regex url_example2 example2\.com
```

Creating a Regular Expression Class Map

A regular expression class map identifies one or more regular expressions. You can use a regular expression class map to match the content of certain traffic; for example, you can match URL strings inside HTTP packets.

To create a regular expression class map, perform the following steps:

Step 1 Create one or more regular expressions according to the "Creating a Regular Expression" section.

Step 2 Create a class map by entering the following command:

```
hostname(config)# class-map type regex match-any class_map_name
hostname(config-cmap)#
```

Where *class_map_name* is a string up to 40 characters in length. The name "class-default" is reserved. All types of class maps use the same name space, so you cannot reuse a name already used by another type of class map.

The **match-any** keyword specifies that the traffic matches the class map if it matches at least one of the regular expressions.

The CLI enters class-map configuration mode.

Step 3 (Optional) Add a description to the class map by entering the following command:

```
hostname(config-cmap)# description string
```

Step 4 Identify the regular expressions you want to include by entering the following command for each regular expression:

```
hostname(config-cmap)# match regex regex_name
```

The following example creates two regular expressions, and adds them to a regular expression class map. Traffic matches the class map if it includes the string "example.com" or "example2.com."

```
hostname(config)# regex url_example example\.com
hostname(config)# regex url_example2 example2\.com
hostname(config)# class-map type regex match-any URLs
hostname(config-cmap)# match regex url_example
hostname(config-cmap)# match regex url_example2
```

Defining Actions (Layer 3/4 Policy Map)

This section describes how to associate actions with Layer 3/4 class maps by creating a Layer 3/4 policy map. This section includes the following topics:

- Information About Layer 3/4 Policy Maps, page 16-17
- Default Layer 3/4 Policy Map, page 16-21
- Adding a Layer 3/4 Policy Map, page 16-22

Information About Layer 3/4 Policy Maps

This section describes how Layer 3/4 policy maps work, and includes the following topics:

- Policy Map Guidelines, page 16-17
- Hierarchical Policy Maps, page 16-17
- Feature Directionality, page 16-18
- Feature Matching Guidelines Within a Policy Map, page 16-18
- Order in Which Multiple Feature Actions are Applied, page 16-19
- Incompatibility of Certain Feature Actions, page 16-20
- Order in Which Multiple Feature Actions are Applied, page 16-19

Policy Map Guidelines

See the following guidelines for using policy maps:

- You can only assign one policy map per interface. (However you can create up to 64 policy maps in the configuration.)
- You can apply the same policy map to multiple interfaces.
- You can identify multiple Layer 3/4 class maps in a Layer 3/4 policy map.
- For each class map, you can assign multiple actions from one or more feature types, if supported. See the "Incompatibility of Certain Feature Actions" section on page 16-20.

Hierarchical Policy Maps

If you enable QoS traffic shaping for a class map, then you can optionally enable priority queueing for a subset of shaped traffic. To do so, you need to create a policy map for the priority queueing, and then within the traffic shaping policy map, you can call the priority class map. Only the traffic shaping class map is applied to an interface.

See Chapter 25, "QoS Overview," for more information about this feature.

Hierarchical policy maps are only supported for traffic shaping and priority queueing.

To implement a hierarchical policy map, perform the following tasks:

1. Identify the prioritized traffic according to the "Identifying Traffic (Layer 3/4 Class Map)" section on page 16-4.

   You can create multiple class maps to be used in the hierarchical policy map.
2. Create a policy map according to the "Defining Actions (Layer 3/4 Policy Map)" section on page 16-16, and identify the sole action for each class map as **priority**.
3. Create a separate policy map according to the "Defining Actions (Layer 3/4 Policy Map)" section on page 16-16, and identify the **shape** action for the **class-default** class map.

   Traffic shaping can only be applied the to **class-default** class map.
4. For the same class map, identify the priority policy map that you created in Step 2 using the **service-policy** *priority_policy_map* command.
5. Apply the shaping policy map to the interface accrding to "Applying Actions to an Interface (Service Policy)" section on page 16-23.

Feature Directionality

Actions are applied to traffic bidirectionally or unidirectionally depending on the feature. For features that are applied bidirectionally, all traffic that enters or exits the interface to which you apply the policy map is affected if the traffic matches the class map for both directions.

Note When you use a global policy, all features are unidirectional; features that are normally bidirectional when applied to a single interface only apply to the ingress of each interface when applied globally. Because the policy is applied to all interfaces, the policy will be applied in both directions so bidirectionality in this case is redundant.

For features that are applied unidirectionally, for example QoS priority queue, only traffic that exits the interface to which you apply the policy map is affected. See Table 16-2 for the directionality of each feature.

Table 16-2 Feature Directionality

Feature	Single Interface Direction	Global Direction
Application inspection (multiple types)	Bidirectional	Ingress
CSC	Bidirectional	Ingress
IPS	Bidirectional	Ingress
QoS input policing	Ingress	Ingress
QoS output policing	Egress	Egress
QoS standard priority queue	Egress	Egress
QoS traffic shaping, hierarchical priority queue	Egress	Egress
TCP normalization, TCP and UDP connection limits and timeouts, and TCP sequence number randomization	Bidirectional	Ingress

Feature Matching Guidelines Within a Policy Map

See the following guidelines for how a packet matches class maps in a policy map:

1. A packet can match only one class map in the policy map for each feature type.
2. When the packet matches a class map for a feature type, the security appliance does not attempt to match it to any subsequent class maps for that feature type.
3. If the packet matches a subsequent class map for a different feature type, however, then the security appliance also applies the actions for the subsequent class map, if supported. See the "Incompatibility of Certain Feature Actions" section on page 16-20 for more information about unsupported combinations.

For example, if a packet matches a class map for connection limits, and also matches a class map for application inspection, then both class map actions are applied.

If a packet matches a class map for HTTP inspection, but also matches another class map that includes HTTP inspection, then the second class map actions are not applied.

Note Application inspection includes multiple inspection types, and each inspection type is a separate feature when you consider the matching guidelines above.

Order in Which Multiple Feature Actions are Applied

The order in which different types of actions in a policy map are performed is independent of the order in which the actions appear in the policy map. Actions are performed in the following order:

1. QoS input policing
2. TCP normalization, TCP and UDP connection limits and timeouts, and TCP sequence number randomization

Note When a the security appliance performs a proxy service (such as AAA or CSC) or it modifies the TCP payload (such as FTP inspection), the TCP normalizer acts in dual mode, where it is applied before and after the proxy or payload modifying service.

3. CSC
4. Application inspection (multiple types)

   The order of application inspections applied when a class of traffic is classified for multiple inspections is as follows. Only one inspection type can be applied to the same traffic. WAAS inspection is an exception, because it can applied along with other inspections for the same traffic. See the "Incompatibility of Certain Feature Actions" section on page 16-20 for more information.

   a. CTIQBE
   b. DNS
   c. FTP
   d. GTP
   e. H323
   f. HTTP
   g. ICMP
   h. ICMP error
   i. ILS
   j. MGCP
   k. NetBIOS
   l. PPTP
   m. Sun RPC
   n. RSH
   o. RTSP
   p. SIP
   q. Skinny
   r. SMTP
   s. SNMP

t. SQL*Net
u. TFTP
v. XDMCP
w. DCERPC
x. Instant Messaging

Note RADIUS accounting is not listed because it is the only inspection allowed on management traffic. WAAS is not listed because it can be configured along with other inspections for the same traffic.

5. IPS
6. QoS output policing
7. QoS standard priority queue
8. QoS traffic shaping, hierarchical priority queue

Incompatibility of Certain Feature Actions

Some features are not compatible with each other for the same traffic. For example, you cannot configure QoS priority queueing and QoS policing for the same set of traffic. Also, most inspections should not be combined with another inspection, so the security appliance only applies one inspection if you configure multiple inspections for the same traffic. In this case, the feature that is applied is the higher priority feature in the list in the "Order in Which Multiple Feature Actions are Applied" section on page 16-19.

For information about compatibility of each feature, see the chapter or section for your feature.

Note The **match default-inspection-traffic** command, which is used in the default global policy, is a special CLI shortcut to match the default ports for all inspections. When used in a policy map, this class map ensures that the correct inspection is applied to each packet, based on the destination port of the traffic. For example, when UDP traffic for port 69 reaches the security appliance, then the security appliance applies the TFTP inspection; when TCP traffic for port 21 arrives, then the security appliance applies the FTP inspection. So in this case only, you can configure multiple inspections for the same class map. Normally, the security appliance does not use the port number to determine the inspection applied, thus giving you the flexibility to apply inspections to non-standard ports, for example.

An example of a misconfiguration is if you configure multiple inspections in the same policy map and do not use the default-inspection-traffic shortcut. In Example 16-1, traffic destined to port 21 is mistakenly configured for both FTP and HTTP inspection. In Example 16-2, traffic destined to port 80 is mistakenly configured for both FTP and HTTP inspection. In both cases of misconfiguration examples, only the FTP inspection is applied, because FTP comes before HTTP in the order of inspections applied.

Example 16-1 Misconfiguration for FTP packets: HTTP Inspection Also Configured

```
class-map ftp
   match port tcp 21
class-map http
   match port tcp 21 [it should be 80]
policy-map test
   class ftp
```

```
    inspect ftp
  class http
    inspect http
```

Example 16-2 Misconfiguration for HTTP packets: FTP Inspection Also Configured

```
class-map ftp
   match port tcp 80 [it should be 21]
class-map http
   match port tcp 80
policy-map test
   class http
     inspect http
   class ftp
     inspect ftp
```

Feature Matching Guidelines for Multiple Policy Maps

For TCP and UDP traffic (and ICMP when you enable stateful ICMP inspection), Modular Policy Framework operates on traffic flows, and not just individual packets. If traffic is part of an existing connection that matches a feature in a policy on one interface, that traffic flow cannot also match the same feature in a policy on another interface; only the first policy is used.

For example, if HTTP traffic matches a policy on the inside interface to inspect HTTP traffic, and you have a separate policy on the outside interface for HTTP inspection, then that traffic is not also inspected on the egress of the outside interface. Similarly, the return traffic for that connection will not be inspected by the ingress policy of the outside interface, nor by the egress policy of the inside interface.

For traffic that is not treated as a flow, for example ICMP when you do not enable stateful ICMP inspection, returning traffic can match a different policy map on the returning interface. For example, if you configure IPS inspection on the inside and outside interfaces, but the inside policy uses virtual sensor 1 while the outside policy uses virtual sensor 2, then a non-stateful Ping will match virtual sensor 1 outbound, but will match virtual sensor 2 inbound.

Default Layer 3/4 Policy Map

The configuration includes a default Layer 3/4 policy map that the security appliance uses in the default global policy. It is called **global_policy** and performs inspection on the default inspection traffic. You can only apply one global policy, so if you want to alter the global policy, you need to either reconfigure the default policy or disable it and apply a new one.

The default policy map configuration includes the following commands:

```
policy-map global_policy
 class inspection_default
  inspect dns preset_dns_map
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect rsh
  inspect rtsp
  inspect esmtp
  inspect sqlnet
  inspect skinny
  inspect sunrpc
  inspect xdmcp
  inspect sip
```

```
inspect netbios
inspect tftp
```

Note See the "Incompatibility of Certain Feature Actions" section on page 16-20 for more information about the special **match default-inspection-traffic** command used in the default class map.

Adding a Layer 3/4 Policy Map

The maximum number of policy maps is 64. To create a Layer 3/4 policy map, perform the following steps:

Step 1 Add the policy map by entering the following command:

```
hostname(config)# policy-map policy_map_name
```

The *policy_map_name* argument is the name of the policy map up to 40 characters in length. All types of policy maps use the same name space, so you cannot reuse a name already used by another type of policy map. The CLI enters policy-map configuration mode.

Step 2 (Optional) Specify a description for the policy map:

```
hostname(config-pmap)# description text
```

Step 3 Specify a previously configured Layer 3/4 class map using the following command:

```
hostname(config-pmap)# class class_map_name
```

where the *class_map_name* is the name of the class map you created earlier. See the "Identifying Traffic (Layer 3/4 Class Map)" section on page 16-4 to add a class map.

Step 4 Specify one or more actions for this class map.

- IPS. See the "Diverting Traffic to the AIP SSM" section on page 23-8.
- CSC. See the "Diverting Traffic to the CSC SSM" section on page 23-16.
- TCP normalization. See the "Configuring TCP Normalization" section on page 24-12.
- TCP and UDP connection limits and timeouts, and TCP sequence number randomization. See the "Configuring Connection Limits and Timeouts" section on page 24-17.
- QoS. See Chapter 25, "Configuring QoS."

  Note You can configure a hierarchical policy map for the traffic shaping and priority queue features. See the "Hierarchical Policy Maps" section on page 16-17 for more information.

- Application inspection. See Chapter 26, "Configuring Application Layer Protocol Inspection."

  Note If there is no **match default_inspection_traffic** command in a class map, then at most one **inspect** command is allowed to be configured under the class.

Step 5 Repeat Step 3 and Step 4 for each class map you want to include in this policy map.

The following is an example of a **policy-map** command for connection policy. It limits the number of connections allowed to the web server 10.1.1.1:

```
hostname(config)# access-list http-server permit tcp any host 10.1.1.1
hostname(config)# class-map http-server
hostname(config-cmap)# match access-list http-server

hostname(config)# policy-map global-policy
hostname(config-pmap)# description This policy map defines a policy concerning connection
to http server.
hostname(config-pmap)# class http-server
hostname(config-pmap-c)# set connection conn-max 256
```

The following example shows how multi-match works in a policy map:

```
hostname(config)# class-map inspection_default
hostname(config-cmap)# match default-inspection-traffic
hostname(config)# class-map http_traffic
hostname(config-cmap)# match port tcp eq 80

hostname(config)# policy-map outside_policy
hostname(config-pmap)# class inspection_default
hostname(config-pmap-c)# inspect http http_map
hostname(config-pmap-c)# inspect sip
hostname(config-pmap)# class http_traffic
hostname(config-pmap-c)# set connection timeout tcp 0:10:0
```

The following example shows how traffic matches the first available class map, and will not match any subsequent class maps that specify actions in the same feature domain:

```
hostname(config)# class-map telnet_traffic
hostname(config-cmap)# match port tcp eq 23
hostname(config)# class-map ftp_traffic
hostname(config-cmap)# match port tcp eq 21
hostname(config)# class-map tcp_traffic
hostname(config-cmap)# match port tcp range 1 65535
hostname(config)# class-map udp_traffic
hostname(config-cmap)# match port udp range 0 65535
hostname(config)# policy-map global_policy
hostname(config-pmap)# class telnet_traffic
hostname(config-pmap-c)# set connection timeout tcp 0:0:0
hostname(config-pmap-c)# set connection conn-max 100
hostname(config-pmap)# class ftp_traffic
hostname(config-pmap-c)# set connection timeout tcp 0:5:0
hostname(config-pmap-c)# set connection conn-max 50
hostname(config-pmap)# class tcp_traffic
hostname(config-pmap-c)# set connection timeout tcp 2:0:0
hostname(config-pmap-c)# set connection conn-max 2000
```

When a Telnet connection is initiated, it matches **class telnet_traffic**. Similarly, if an FTP connection is initiated, it matches **class ftp_traffic**. For any TCP connection other than Telnet and FTP, it will match **class tcp_traffic**. Even though a Telnet or FTP connection can match **class tcp_traffic**, the security appliance does not make this match because they previously matched other classes.

Applying Actions to an Interface (Service Policy)

To activate the Layer 3/4 policy map, create a service policy that applies it to one or more interfaces or that applies it globally to all interfaces. Interface service policies take precedence over the global service policy for a given feature. For example, if you have a global policy with FTP inspection, and an interface

policy with TCP normalization, then both FTP inspection and TCP normalization are applied to the interface. However, if you have a global policy with FTP inspection, and an interface policy with FTP inspection, then only the interface policy FTP inspection is applied to that interface.

- To create a service policy by associating a policy map with an interface, enter the following command:

  ```
  hostname(config)# service-policy policy_map_name interface interface_name
  ```

- To create a service policy that applies to all interfaces that do not have a specific policy, enter the following command:

  ```
  hostname(config)# service-policy policy_map_name global
  ```

  By default, the configuration includes a global policy that matches all default application inspection traffic and applies inspection to the traffic globally. You can only apply one global policy, so if you want to alter the global policy, you need to either edit the default policy or disable it and apply a new one.

  The default service policy includes the following command:

  ```
  service-policy global_policy global
  ```

For example, the following command enables the inbound_policy policy map on the outside interface:

```
hostname(config)# service-policy inbound_policy interface outside
```

The following commands disable the default global policy, and enables a new one called new_global_policy on all other security appliance interfaces:

```
hostname(config)# no service-policy global_policy global
hostname(config)# service-policy new_global_policy global
```

Modular Policy Framework Examples

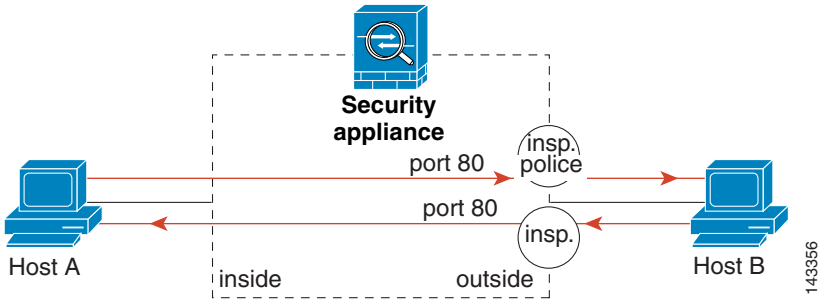
This section includes several Modular Policy Framework examples, and includes the following topics:

- Applying Inspection and QoS Policing to HTTP Traffic, page 16-25
- Applying Inspection to HTTP Traffic Globally, page 16-25
- Applying Inspection and Connection Limits to HTTP Traffic to Specific Servers, page 16-26
- Applying Inspection to HTTP Traffic with NAT, page 16-27

Applying Inspection and QoS Policing to HTTP Traffic

In this example (see Figure 16-1), any HTTP connection (TCP traffic on port 80) that enters or exits the security appliance through the outside interface is classified for HTTP inspection. Any HTTP traffic that exits the outside interface is classified for policing.

Figure 16-1 HTTP Inspection and QoS Policing



See the following commands for this example:

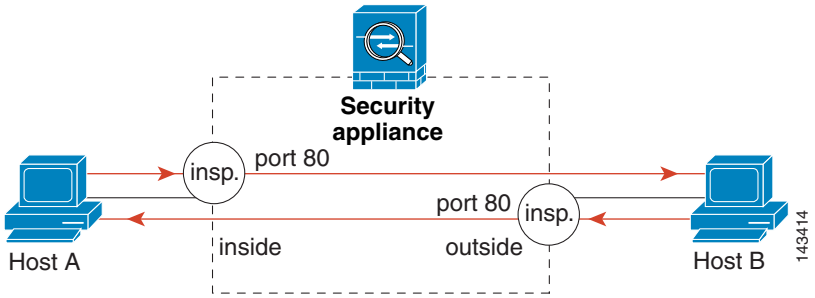
```
hostname(config)# class-map http_traffic
hostname(config-cmap)# match port tcp eq 80

hostname(config)# policy-map http_traffic_policy
hostname(config-pmap)# class http_traffic
hostname(config-pmap-c)# inspect http
hostname(config-pmap-c)# police output 250000
hostname(config)# service-policy http_traffic_policy interface outside
```

Applying Inspection to HTTP Traffic Globally

In this example (see Figure 16-2), any HTTP connection (TCP traffic on port 80) that enters the security appliance through any interface is classified for HTTP inspection. Because the policy is a global policy, inspection occurs only as the traffic enters each interface.

Figure 16-2 Global HTTP Inspection



See the following commands for this example:

```
hostname(config)# class-map http_traffic
hostname(config-cmap)# match port tcp eq 80
```

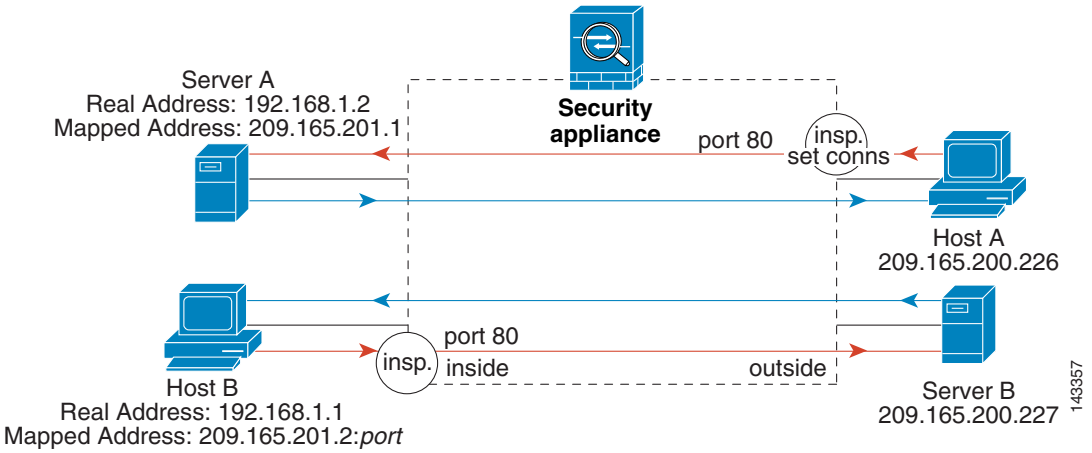
```
hostname(config)# policy-map http_traffic_policy
hostname(config-pmap)# class http_traffic
hostname(config-pmap-c)# inspect http
hostname(config)# service-policy http_traffic_policy global
```

Applying Inspection and Connection Limits to HTTP Traffic to Specific Servers

In this example (see Figure 16-3), any HTTP connection destined for Server A (TCP traffic on port 80) that enters the security appliance through the outside interface is classified for HTTP inspection and maximum connection limits. Connections initiated from server A to Host A does not match the access list in the class map, so it is not affected.

Any HTTP connection destined for Server B that enters the security appliance through the inside interface is classified for HTTP inspection. Connections initiated from server B to Host B does not match the access list in the class map, so it is not affected.

Figure 16-3 HTTP Inspection and Connection Limits to Specific Servers



See the following commands for this example:

```
hostname(config)# static (inside,outside) 209.165.201.1 192.168.1.2
hostname(config)# nat (inside) 1 192.168.1.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.2
hostname(config)# access-list serverA extended permit tcp any host 209.165.201.1 eq 80
hostname(config)# access-list ServerB extended permit tcp any host 209.165.200.227 eq 80

hostname(config)# class-map http_serverA
hostname(config-cmap)# match access-list serverA
hostname(config)# class-map http_serverB
hostname(config-cmap)# match access-list serverB

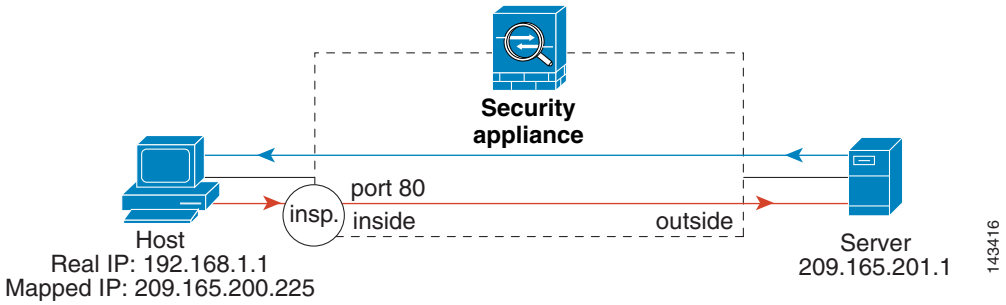
hostname(config)# policy-map policy_serverA
hostname(config-pmap)# class http_serverA
hostname(config-pmap-c)# inspect http
hostname(config-pmap-c)# set connection conn-max 100
hostname(config)# policy-map policy_serverB
hostname(config-pmap)# class http_serverB
hostname(config-pmap-c)# inspect http

hostname(config)# service-policy policy_serverB interface inside
hostname(config)# service-policy policy_serverA interface outside
```

Applying Inspection to HTTP Traffic with NAT

In this example, the Host on the inside network has two addresses: one is the real IP address 192.168.1.1, and the other is a mapped IP address used on the outside network, 209.165.200.225. Because the policy is applied to the inside interface, where the real address is used, then you must use the real IP address in the access list in the class map. If you applied it to the outside interface, you would use the mapped address.

Figure 16-4 HTTP Inspection with NAT



See the following commands for this example:

```
hostname(config)# static (inside,outside) 209.165.200.225 192.168.1.1
hostname(config)# access-list http_client extended permit tcp host 192.168.1.1 any eq 80

hostname(config)# class-map http_client
hostname(config-cmap)# match access-list http_client

hostname(config)# policy-map http_client
hostname(config-pmap)# class http_client
hostname(config-pmap-c)# inspect http

hostname(config)# service-policy http_client interface inside
```



PART 2

Configuring the Firewall



CHAPTER 17

Firewall Mode Overview

This chapter describes how the firewall works in each firewall mode. To set the firewall mode, see the "Setting Transparent or Routed Firewall Mode" section on page 2-5.

Note In multiple context mode, you cannot set the firewall mode separately for each context; you can only set the firewall mode for the entire security appliance.

This chapter includes the following sections:

- Routed Mode Overview, page 17-1
- Transparent Mode Overview, page 17-7

Routed Mode Overview

In routed mode, the security appliance is considered to be a router hop in the network. It can use OSPF or RIP (in single context mode). Routed mode supports many interfaces. Each interface is on a different subnet. You can share interfaces between contexts.

This section includes the following topics:

- IP Routing Support, page 17-1
- How Data Moves Through the Security Appliance in Routed Firewall Mode, page 17-1

IP Routing Support

The security appliance acts as a router between connected networks, and each interface requires an IP address on a different subnet. In single context mode, the routed firewall supports OSPF and RIP. Multiple context mode supports static routes only. We recommend using the advanced routing capabilities of the upstream and downstream routers instead of relying on the security appliance for extensive routing needs.

How Data Moves Through the Security Appliance in Routed Firewall Mode

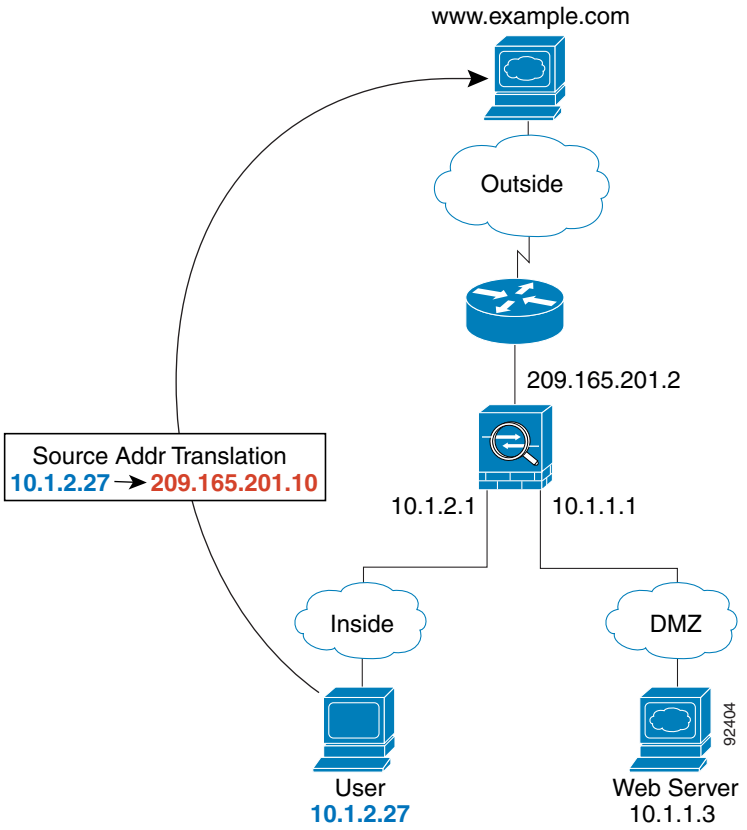
This section describes how data moves through the security appliance in routed firewall mode, and includes the following topics:

- An Inside User Visits a Web Server, page 17-2
- An Outside User Visits a Web Server on the DMZ, page 17-3
- An Inside User Visits a Web Server on the DMZ, page 17-4
- An Outside User Attempts to Access an Inside Host, page 17-5
- A DMZ User Attempts to Access an Inside Host, page 17-6

An Inside User Visits a Web Server

Figure 17-1 shows an inside user accessing an outside web server.

Figure 17-1 Inside to Outside



The following steps describe how data moves through the security appliance (see Figure 17-1):

1. The user on the inside network requests a web page from www.example.com.
2. The security appliance receives the packet and because it is a new session, the security appliance verifies that the packet is allowed according to the terms of the security policy (access lists, filters, AAA).

   For multiple context mode, the security appliance first classifies the packet according to either a unique interface or a unique destination address associated with a context; the destination address is associated by matching an address translation in a context. In this case, the interface would be unique; the www.example.com IP address does not have a current address translation in a context.

3. The security appliance translates the local source address (10.1.2.27) to the global address 209.165.201.10, which is on the outside interface subnet.

   The global address could be on any subnet, but routing is simplified when it is on the outside interface subnet.

4. The security appliance then records that a session is established and forwards the packet from the outside interface.

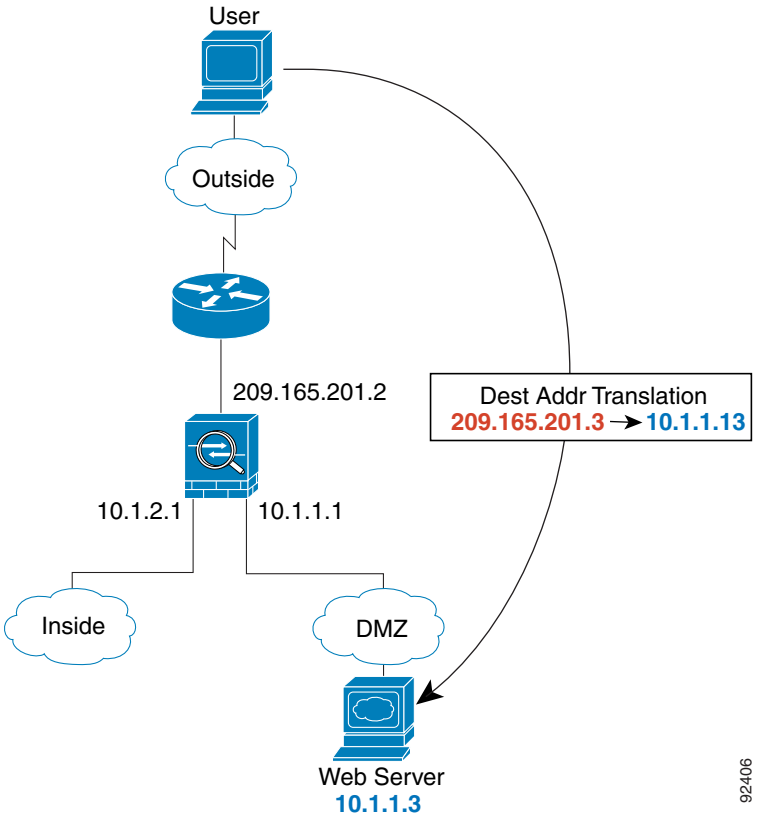
5. When www.example.com responds to the request, the packet goes through the security appliance, and because the session is already established, the packet bypasses the many lookups associated with a new connection. The security appliance performs NAT by translating the global destination address to the local user address, 10.1.2.27.

6. The security appliance forwards the packet to the inside user.

An Outside User Visits a Web Server on the DMZ

Figure 17-2 shows an outside user accessing the DMZ web server.

Figure 17-2 Outside to DMZ



The following steps describe how data moves through the security appliance (see Figure 17-2):

1. A user on the outside network requests a web page from the DMZ web server using the global destination address of 209.165.201.3, which is on the outside interface subnet.

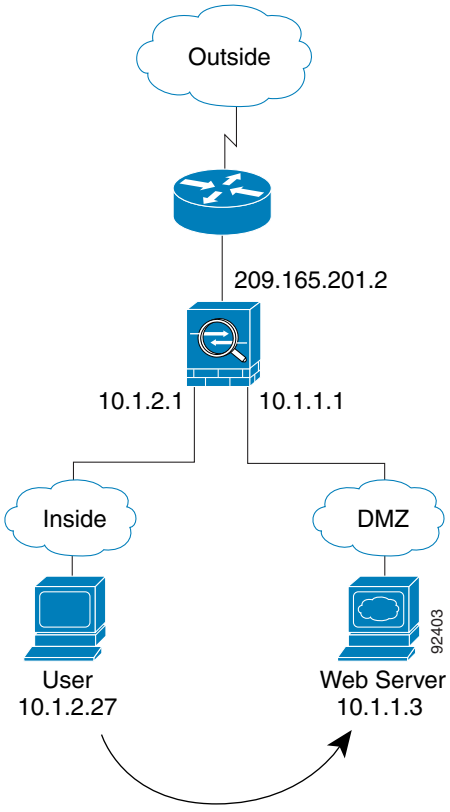
2. The security appliance receives the packet and because it is a new session, the security appliance verifies that the packet is allowed according to the terms of the security policy (access lists, filters, AAA).

   For multiple context mode, the security appliance first classifies the packet according to either a unique interface or a unique destination address associated with a context; the destination address is associated by matching an address translation in a context. In this case, the classifier "knows" that the DMZ web server address belongs to a certain context because of the server address translation.
3. The security appliance translates the destination address to the local address 10.1.1.3.
4. The security appliance then adds a session entry to the fast path and forwards the packet from the DMZ interface.
5. When the DMZ web server responds to the request, the packet goes through the security appliance and because the session is already established, the packet bypasses the many lookups associated with a new connection. The security appliance performs NAT by translating the local source address to 209.165.201.3.
6. The security appliance forwards the packet to the outside user.

An Inside User Visits a Web Server on the DMZ

Figure 17-3 shows an inside user accessing the DMZ web server.

Figure 17-3 Inside to DMZ



The following steps describe how data moves through the security appliance (see Figure 17-3):

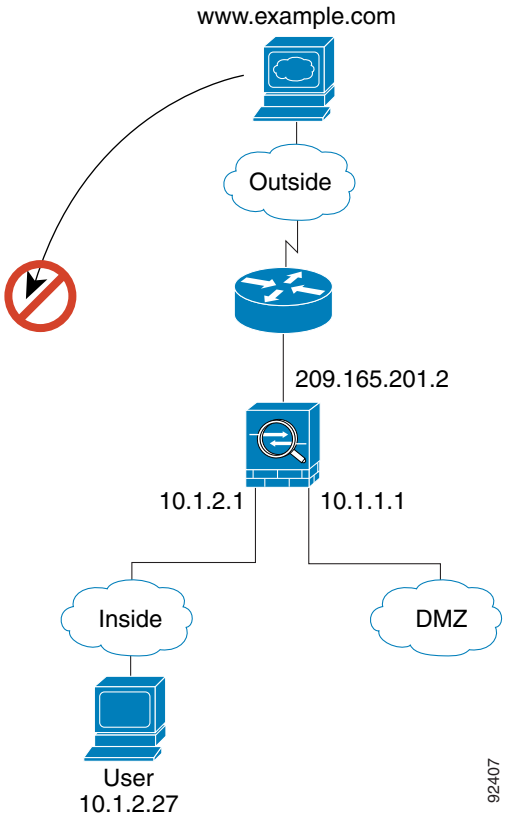
1. A user on the inside network requests a web page from the DMZ web server using the destination address of 10.1.1.3.
2. The security appliance receives the packet and because it is a new session, the security appliance verifies that the packet is allowed according to the terms of the security policy (access lists, filters, AAA).

   For multiple context mode, the security appliance first classifies the packet according to either a unique interface or a unique destination address associated with a context; the destination address is associated by matching an address translation in a context. In this case, the interface is unique; the web server IP address does not have a current address translation.
3. The security appliance then records that a session is established and forwards the packet out of the DMZ interface.
4. When the DMZ web server responds to the request, the packet goes through the fast path, which lets the packet bypass the many lookups associated with a new connection.
5. The security appliance forwards the packet to the inside user.

An Outside User Attempts to Access an Inside Host

Figure 17-4 shows an outside user attempting to access the inside network.

Figure 17-4 Outside to Inside



The following steps describe how data moves through the security appliance (see Figure 17-4):

1. A user on the outside network attempts to reach an inside host (assuming the host has a routable IP address).

   If the inside network uses private addresses, no outside user can reach the inside network without NAT. The outside user might attempt to reach an inside user by using an existing NAT session.

2. The security appliance receives the packet and because it is a new session, the security appliance verifies if the packet is allowed according to the security policy (access lists, filters, AAA).

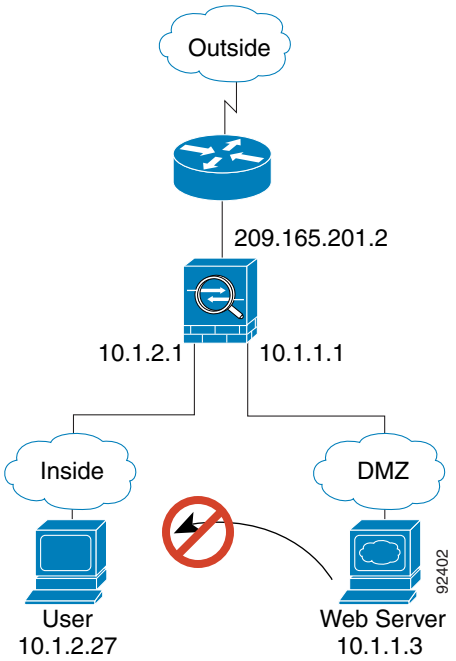
3. The packet is denied, and the security appliance drops the packet and logs the connection attempt.

   If the outside user is attempting to attack the inside network, the security appliance employs many technologies to determine if a packet is valid for an already established session.

A DMZ User Attempts to Access an Inside Host

Figure 17-5 shows a user in the DMZ attempting to access the inside network.

Figure 17-5 DMZ to Inside



The following steps describe how data moves through the security appliance (see Figure 17-5):

1. A user on the DMZ network attempts to reach an inside host. Because the DMZ does not have to route the traffic on the Internet, the private addressing scheme does not prevent routing.

2. The security appliance receives the packet and because it is a new session, the security appliance verifies if the packet is allowed according to the security policy (access lists, filters, AAA).

3. The packet is denied, and the security appliance drops the packet and logs the connection attempt.

Transparent Mode Overview

Traditionally, a firewall is a routed hop and acts as a default gateway for hosts that connect to one of its screened subnets. A transparent firewall, on the other hand, is a Layer 2 firewall that acts like a "bump in the wire," or a "stealth firewall," and is not seen as a router hop to connected devices.

This section describes transparent firewall mode, and includes the following topics:

- Transparent Firewall Network, page 17-7
- Allowing Layer 3 Traffic, page 17-7
- Allowed MAC Addresses, page 17-7
- Passing Traffic Not Allowed in Routed Mode, page 17-8
- MAC Address vs. Route Lookups, page 17-8
- Using the Transparent Firewall in Your Network, page 17-9
- Transparent Firewall Guidelines, page 17-9
- Unsupported Features in Transparent Mode, page 17-10
- How Data Moves Through the Transparent Firewall, page 17-11

Transparent Firewall Network

The security appliance connects the same network on its inside and outside interfaces. Because the firewall is not a routed hop, you can easily introduce a transparent firewall into an existing network.

Allowing Layer 3 Traffic

IPv4 traffic is allowed through the transparent firewall automatically from a higher security interface to a lower security interface, without an access list. ARPs are allowed through the transparent firewall in both directions without an access list. ARP traffic can be controlled by ARP inspection. For Layer 3 traffic travelling from a low to a high security interface, an extended access list is required on the low security interface. See the "Adding an Extended Access List" section on page 18-5 for more information.

Allowed MAC Addresses

The following destination MAC addresses are allowed through the transparent firewall. Any MAC address not on this list is dropped.

- TRUE broadcast destination MAC address equal to FFFF.FFFF.FFFF
- IPv4 multicast MAC addresses from 0100.5E00.0000 to 0100.5EFE.FFFF
- IPv6 multicast MAC addresses from 3333.0000.0000 to 3333.FFFF.FFFF
- BPDU multicast address equal to 0100.0CCC.CCCD
- Appletalk multicast MAC addresses from 0900.0700.0000 to 0900.07FF.FFFF

Passing Traffic Not Allowed in Routed Mode

In routed mode, some types of traffic cannot pass through the security appliance even if you allow it in an access list. The transparent firewall, however, can allow almost any traffic through using either an extended access list (for IP traffic) or an EtherType access list (for non-IP traffic).

Note The transparent mode security appliance does not pass CDP packets or IPv6 packets, or any packets that do not have a valid EtherType greater than or equal to 0x600. For example, you cannot pass IS-IS packets. An exception is made for BPDUs, which are supported.

For example, you can establish routing protocol adjacencies through a transparent firewall; you can allow OSPF, RIP, EIGRP, or BGP traffic through based on an extended access list. Likewise, protocols like HSRP or VRRP can pass through the security appliance.

Non-IP traffic (for example AppleTalk, IPX, BPDUs, and MPLS) can be configured to go through using an EtherType access list.

For features that are not directly supported on the transparent firewall, you can allow traffic to pass through so that upstream and downstream routers can support the functionality. For example, by using an extended access list, you can allow DHCP traffic (instead of the unsupported DHCP relay feature) or multicast traffic such as that created by IP/TV.

MAC Address vs. Route Lookups

When the security appliance runs in transparent mode without NAT, the outgoing interface of a packet is determined by performing a MAC address lookup instead of a route lookup. Route statements can still be configured, but they only apply to security appliance-originated traffic. For example, if your syslog server is located on a remote network, you must use a static route so the security appliance can reach that subnet.

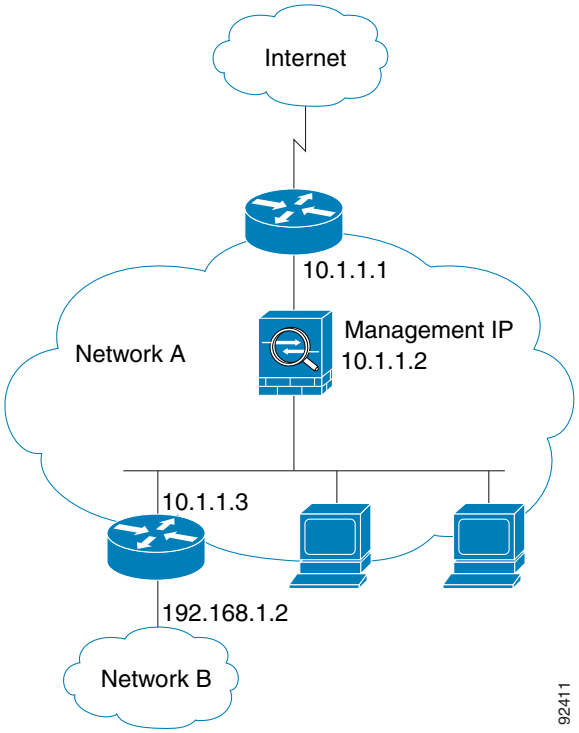
An exception to this rule is when you use voice inspections and the endpoint is at least one hop away from the security appliance. For example, if you use the transparent firewall between a CCM and an H.323 gateway, and there is a router between the transparent firewall and the H.323 gateway, then you need to add a static route on the security appliance for the H.323 gateway for successful call completion.

If you use NAT, then the security appliance uses a route lookup instead of a MAC address lookup. In some cases, you will need static routes. For example, if the real destination address is not directly-connected to the security appliance, then you need to add a static route on the security appliance for the real destination address that points to the downstream router.

Using the Transparent Firewall in Your Network

Figure 17-6 shows a typical transparent firewall network where the outside devices are on the same subnet as the inside devices. The inside router and hosts appear to be directly connected to the outside router.

Figure 17-6 Transparent Firewall Network



Transparent Firewall Guidelines

Follow these guidelines when planning your transparent firewall network:

- A management IP address is required; for multiple context mode, an IP address is required for each context.

  Unlike routed mode, which requires an IP address for each interface, a transparent firewall has an IP address assigned to the entire device. The security appliance uses this IP address as the source address for packets originating on the security appliance, such as system messages or AAA communications.

  The management IP address must be on the same subnet as the connected network. You cannot set the subnet to a host subnet (255.255.255.255).

  You can configure an IP address for the Management 0/0 management-only interface. This IP address can be on a separate subnet from the main management IP address.

Note If the management IP address is not configured, transient traffic does not pass through the transparent firewall. For multiple context mode, transient traffic does not pass through virtual contexts.

- The transparent security appliance uses an inside interface and an outside interface only. If your platform includes a dedicated management interface, you can also configure the management interface or subinterface for management traffic only.

  In single mode, you can only use two data interfaces (and the dedicated management interface, if available) even if your security appliance includes more than two interfaces.
- Each directly connected network must be on the same subnet.
- Do not specify the security appliance management IP address as the default gateway for connected devices; devices need to specify the router on the other side of the security appliance as the default gateway.
- For multiple context mode, each context must use different interfaces; you cannot share an interface across contexts.
- For multiple context mode, each context typically uses a different subnet. You can use overlapping subnets, but your network topology requires router and NAT configuration to make it possible from a routing standpoint.

Unsupported Features in Transparent Mode

Table 17-1 lists the features are not supported in transparent mode.

Table 17-1 Unsupported Features in Transparent Mode

Feature	Description
Dynamic DNS	—
DHCP relay	The transparent firewall can act as a DHCP server, but it does not support the DHCP relay commands. DHCP relay is not required because you can allow DHCP traffic to pass through using two extended access lists: one that allows DCHP requests from the inside interface to the outside, and one that allows the replies from the server in the other direction.
Dynamic routing protocols	You can, however, add static routes for traffic originating on the security appliance. You can also allow dynamic routing protocols through the security appliance using an extended access list.
IPv6	You also cannot allow IPv6 using an EtherType access list.
Multicast	You can allow multicast traffic through the security appliance by allowing it in an extended access list.

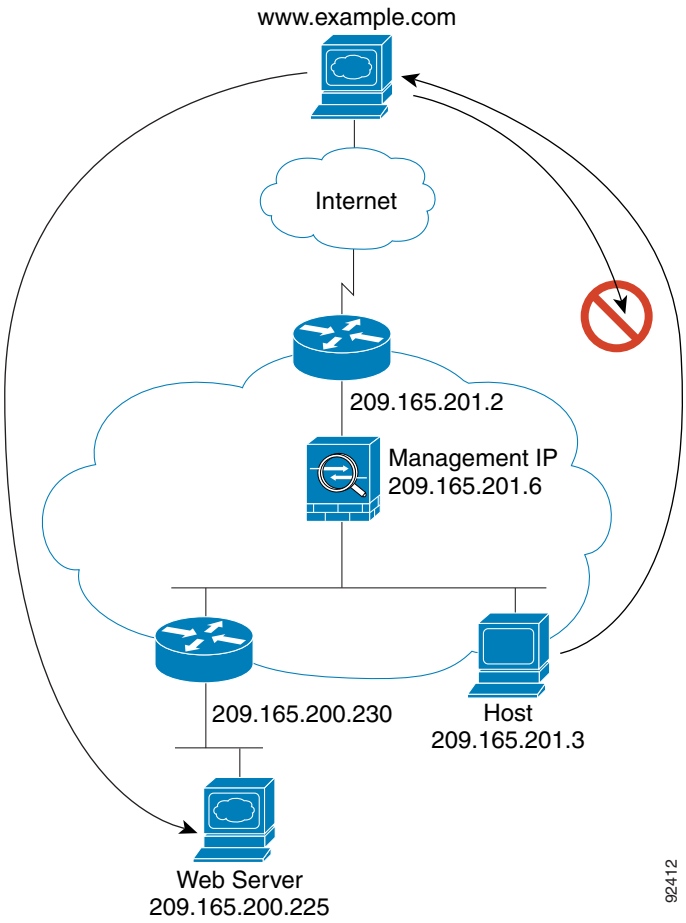
Table 17-1 Unsupported Features in Transparent Mode (continued)

Feature	Description
QoS	—
VPN termination for through traffic	The transparent firewall supports site-to-site VPN tunnels for management connections only. It does not terminate VPN connections for traffic through the security appliance. You can pass VPN traffic through the security appliance using an extended access list, but it does not terminate non-management connections. WebVPN is also not supported.

How Data Moves Through the Transparent Firewall

Figure 17-7 shows a typical transparent firewall implementation with an inside network that contains a public web server. The security appliance has an access list so that the inside users can access Internet resources. Another access list lets the outside users access only the web server on the inside network.

Figure 17-7 Typical Transparent Firewall Data Path



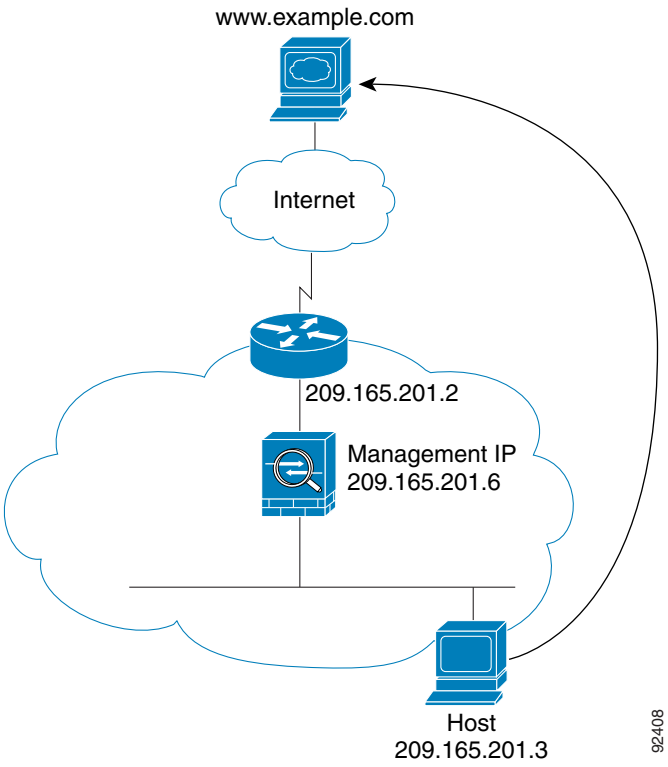
This section describes how data moves through the security appliance, and includes the following topics:

- An Inside User Visits a Web Server, page 17-12
- An Inside User Visits a Web Server Using NAT, page 17-13
- An Outside User Visits a Web Server on the Inside Network, page 17-14
- An Outside User Attempts to Access an Inside Host, page 17-15

An Inside User Visits a Web Server

Figure 17-8 shows an inside user accessing an outside web server.

Figure 17-8 Inside to Outside



The following steps describe how data moves through the security appliance (see Figure 17-8):

1. The user on the inside network requests a web page from www.example.com.
2. The security appliance receives the packet and adds the source MAC address to the MAC address table, if required. Because it is a new session, it verifies that the packet is allowed according to the terms of the security policy (access lists, filters, AAA).

   For multiple context mode, the security appliance first classifies the packet according to a unique interface.
3. The security appliance records that a session is established.
4. If the destination MAC address is in its table, the security appliance forwards the packet out of the outside interface. The destination MAC address is that of the upstream router, 209.186.201.2.

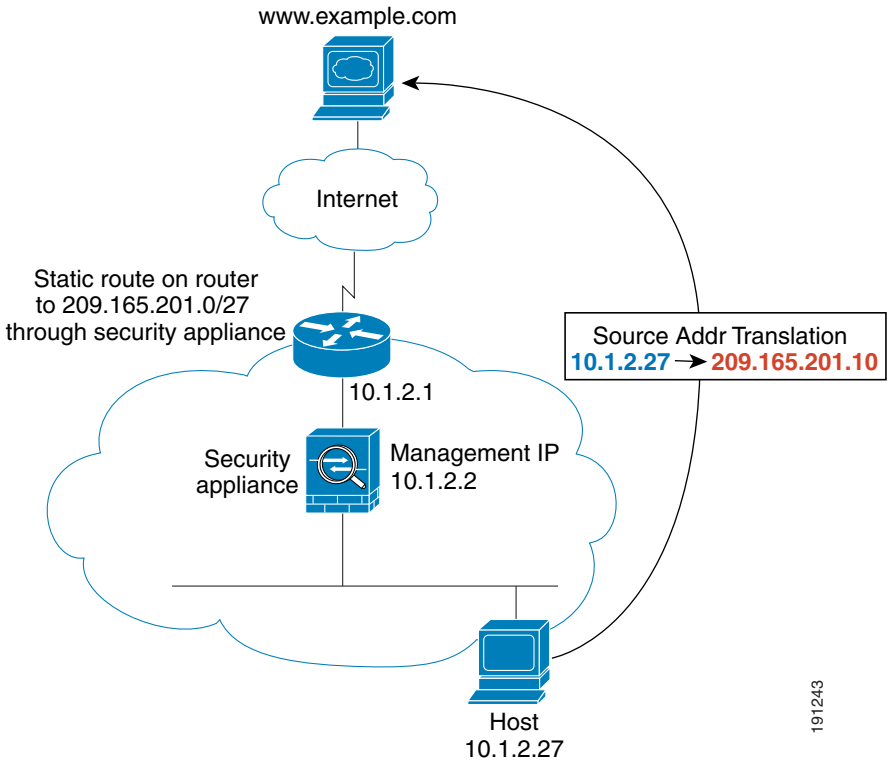
If the destination MAC address is not in the security appliance table, the security appliance attempts to discover the MAC address by sending an ARP request and a ping. The first packet is dropped.

5. The web server responds to the request; because the session is already established, the packet bypasses the many lookups associated with a new connection.
6. The security appliance forwards the packet to the inside user.

An Inside User Visits a Web Server Using NAT

Figure 17-8 shows an inside user accessing an outside web server.

Figure 17-9 Inside to Outside with NAT



The following steps describe how data moves through the security appliance (see Figure 17-8):

1. The user on the inside network requests a web page from www.example.com.
2. The security appliance receives the packet and adds the source MAC address to the MAC address table, if required. Because it is a new session, it verifies that the packet is allowed according to the terms of the security policy (access lists, filters, AAA).

   For multiple context mode, the security appliance first classifies the packet according to a unique interface.
3. The security appliance translates the real address (10.1.2.27) to the mapped address 209.165.201.10.

   Because the mapped address is not on the same network as the outside interface, then be sure the upstream router has a static route to the mapped network that points to the security appliance.
4. The security appliance then records that a session is established and forwards the packet from the outside interface.

5. If the destination MAC address is in its table, the security appliance forwards the packet out of the outside interface. The destination MAC address is that of the upstream router, 209.165.201.2.

   If the destination MAC address is not in the security appliance table, the security appliance attempts to discover the MAC address by sending an ARP request and a ping. The first packet is dropped.

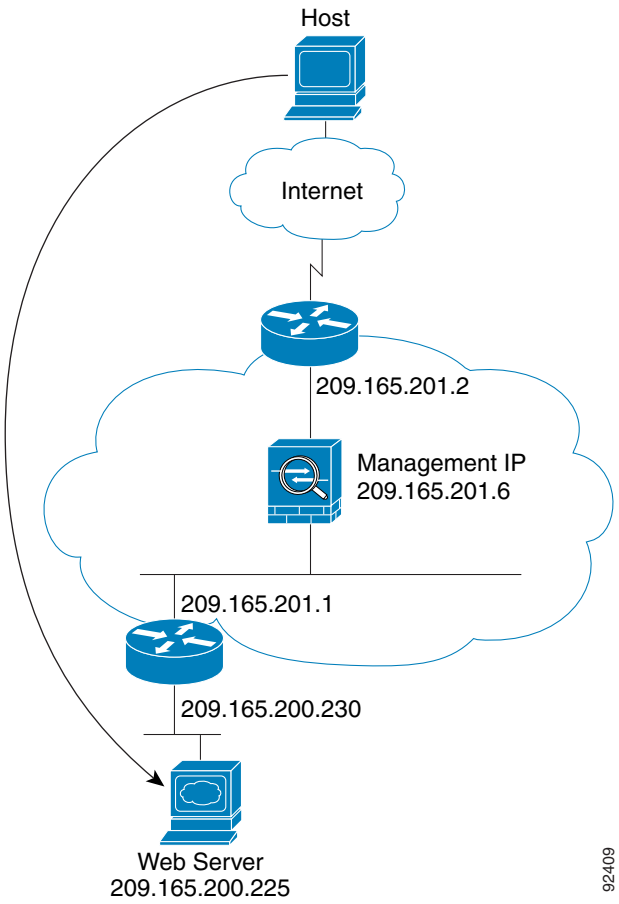
6. The web server responds to the request; because the session is already established, the packet bypasses the many lookups associated with a new connection.

7. The security appliance performs NAT by translating the mapped address to the real address, 10.1.2.27.

An Outside User Visits a Web Server on the Inside Network

Figure 17-10 shows an outside user accessing the inside web server.

Figure 17-10 Outside to Inside



The following steps describe how data moves through the security appliance (see Figure 17-10):

1. A user on the outside network requests a web page from the inside web server.

2. The security appliance receives the packet and adds the source MAC address to the MAC address table, if required. Because it is a new session, it verifies that the packet is allowed according to the terms of the security policy (access lists, filters, AAA).

For multiple context mode, the security appliance first classifies the packet according to a unique interface.

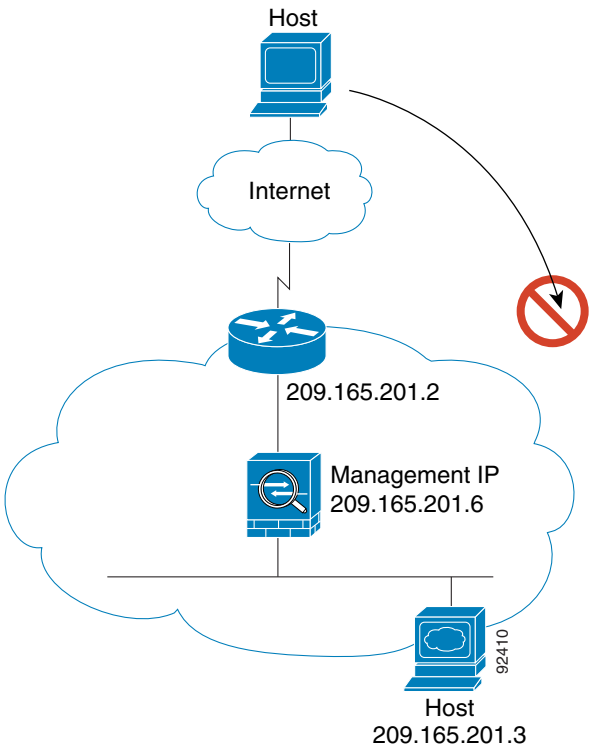
3. The security appliance records that a session is established.
4. If the destination MAC address is in its table, the security appliance forwards the packet out of the inside interface. The destination MAC address is that of the downstream router, 209.186.201.1.

   If the destination MAC address is not in the security appliance table, the security appliance attempts to discover the MAC address by sending an ARP request and a ping. The first packet is dropped.
5. The web server responds to the request; because the session is already established, the packet bypasses the many lookups associated with a new connection.
6. The security appliance forwards the packet to the outside user.

An Outside User Attempts to Access an Inside Host

Figure 17-11 shows an outside user attempting to access a host on the inside network.

Figure 17-11 Outside to Inside



The following steps describe how data moves through the security appliance (see Figure 17-11):

1. A user on the outside network attempts to reach an inside host.
2. The security appliance receives the packet and adds the source MAC address to the MAC address table, if required. Because it is a new session, it verifies if the packet is allowed according to the terms of the security policy (access lists, filters, AAA).

   For multiple context mode, the security appliance first classifies the packet according to a unique interface.

3. The packet is denied, and the security appliance drops the packet.
4. If the outside user is attempting to attack the inside network, the security appliance employs many technologies to determine if a packet is valid for an already established session.



CHAPTER 18

Identifying Traffic with Access Lists

This chapter describes how to identify traffic with access lists. This chapter includes the following topics:

- Access List Overview, page 18-1
- Adding an Extended Access List, page 18-5
- Adding an EtherType Access List, page 18-8
- Adding a Standard Access List, page 18-11
- Adding a Webtype Access List, page 18-11
- Simplifying Access Lists with Object Grouping, page 18-12
- Adding Remarks to Access Lists, page 18-19
- Scheduling Extended Access List Activation, page 18-19
- Logging Access List Activity, page 18-21

For information about IPv6 access lists, see the "Configuring IPv6 Access Lists" section on page 13-6.

Access List Overview

Access lists are made up of one or more Access Control Entries. An ACE is a single entry in an access list that specifies a permit or deny rule, and is applied to a protocol, a source and destination IP address or network, and optionally the source and destination ports.

Access lists are used in a variety of features. If your feature uses Modular Policy Framework, you can use an access list to identify traffic within a traffic class map. For more information on Modular Policy Framework, see Chapter 16, "Using Modular Policy Framework."

This section includes the following topics:

- Access List Types, page 18-2
- Access Control Entry Order, page 18-2
- Access Control Implicit Deny, page 18-3
- IP Addresses Used for Access Lists When You Use NAT, page 18-3

Access List Types

Table 18-1 lists the types of access lists and some common uses for them.

Table 18-1 Access List Types and Common Uses

Access List Use	Access List Type	Description
Control network access for IP traffic (routed and transparent mode)	Extended	The security appliance does not allow any traffic from a lower security interface to a higher security interface unless it is explicitly permitted by an extended access list. **Note** To access the security appliance interface for management access, you do not also need an access list allowing the host IP address. You only need to configure management access according to Chapter 42, "Managing System Access."
Identify traffic for AAA rules	Extended	AAA rules use access lists to identify traffic.
Control network access for IP traffic for a given user	Extended, downloaded from a AAA server per user	You can configure the RADIUS server to download a dynamic access list to be applied to the user, or the server can send the name of an access list that you already configured on the security appliance.
Identify addresses for NAT (policy NAT and NAT exemption)	Extended	Policy NAT lets you identify local traffic for address translation by specifying the source and destination addresses in an extended access list.
Establish VPN access	Extended	You can use an extended access list in VPN commands.
Identify traffic in a traffic class map for Modular Policy Framework	Extended EtherType	Access lists can be used to identify traffic in a class map, which is used for features that support Modular Policy Framework. Features that support Modular Policy Framework include TCP and general connection settings, and inspection.
For transparent firewall mode, control network access for non-IP traffic	EtherType	You can configure an access list that controls traffic based on its EtherType.
Identify OSPF route redistribution	Standard	Standard access lists include only the destination address. You can use a standard access list to control the redistribution of OSPF routes.
Filtering for WebVPN	Webtype	You can configure a Webtype access list to filter URLs.

Access Control Entry Order

An access list is made up of one or more Access Control Entries. Depending on the access list type, you can specify the source and destination addresses, the protocol, the ports (for TCP or UDP), the ICMP type (for ICMP), or the EtherType.

Each ACE that you enter for a given access list name is appended to the end of the access list.

The order of ACEs is important. When the security appliance decides whether to forward or drop a packet, the security appliance tests the packet against each ACE in the order in which the entries are listed. After a match is found, no more ACEs are checked. For example, if you create an ACE at the beginning of an access list that explicitly permits all traffic, no further statements are ever checked.

You can disable an ACE by specifying the keyword **inactive** in the **access-list** command.

Access Control Implicit Deny

Access lists have an implicit deny at the end of the list, so unless you explicitly permit it, traffic cannot pass. For example, if you want to allow all users to access a network through the security appliance except for particular addresses, then you need to deny the particular addresses and then permit all others.

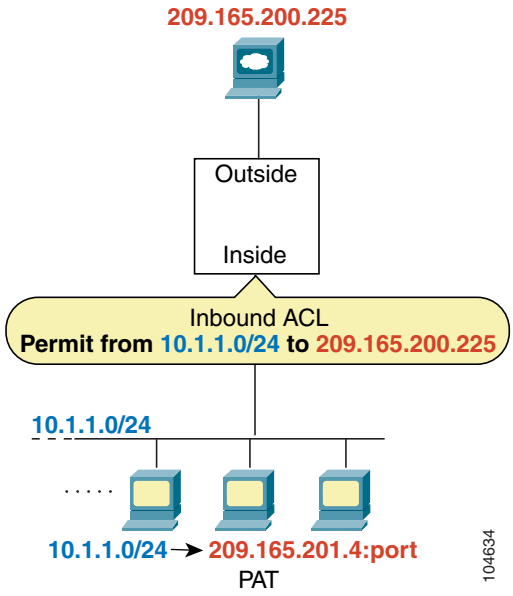
For EtherType access lists, the implicit deny at the end of the access list does not affect IP traffic or ARPs; for example, if you allow EtherType 8037, the implicit deny at the end of the access list does not now block any IP traffic that you previously allowed with an extended access list (or implicitly allowed from a high security interface to a low security interface). However, if you *explicitly* deny all traffic with an EtherType ACE, then IP and ARP traffic is denied.

IP Addresses Used for Access Lists When You Use NAT

When you use NAT, the IP addresses you specify for an access list depend on the interface to which the access list is attached; you need to use addresses that are valid on the network connected to the interface. This guideline applies for both inbound and outbound access lists: the direction does not determine the address used, only the interface does.

For example, you want to apply an access list to the inbound direction of the inside interface. You configure the security appliance to perform NAT on the inside source addresses when they access outside addresses. Because the access list is applied to the inside interface, the source addresses are the original untranslated addresses. Because the outside addresses are not translated, the destination address used in the access list is the real address (see Figure 18-1).

Figure 18-1 IP Addresses in Access Lists: NAT Used for Source Addresses



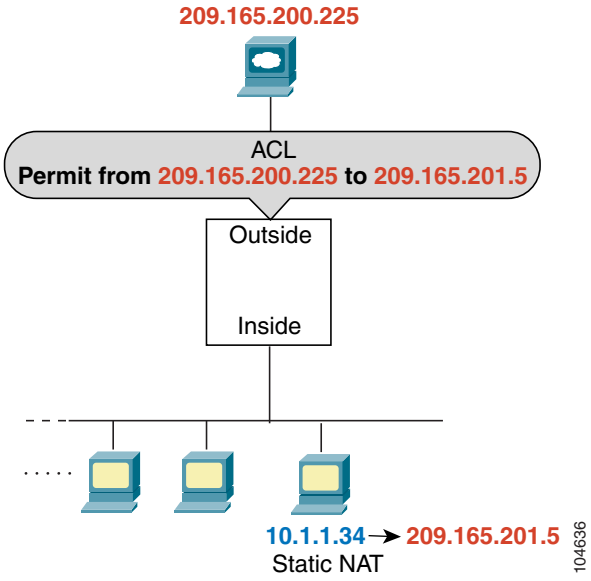
See the following commands for this example:

```
hostname(config)# access-list INSIDE extended permit ip 10.1.1.0 255.255.255.0 host
209.165.200.225
```

```
hostname(config)# access-group INSIDE in interface inside
```

If you want to allow an outside host to access an inside host, you can apply an inbound access list on the outside interface. You need to specify the translated address of the inside host in the access list because that address is the address that can be used on the outside network (see Figure 18-2).

Figure 18-2 IP Addresses in Access Lists: NAT used for Destination Addresses

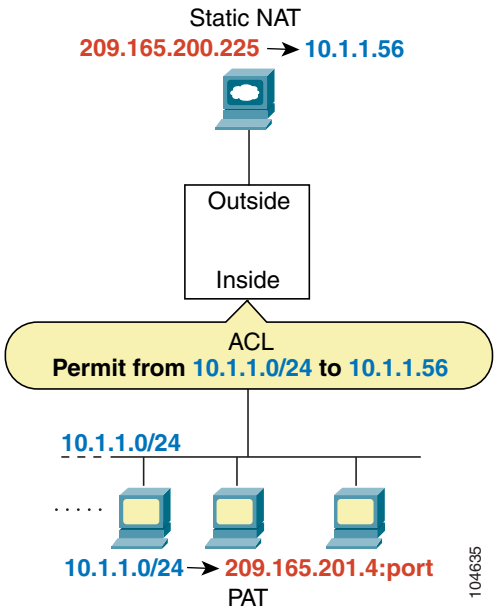


See the following commands for this example:

```
hostname(config)# access-list OUTSIDE extended permit ip host 209.165.200.225 host
209.165.201.5
hostname(config)# access-group OUTSIDE in interface outside
```

If you perform NAT on both interfaces, keep in mind the addresses that are visible to a given interface. In Figure 18-3, an outside server uses static NAT so that a translated address appears on the inside network.

Figure 18-3 IP Addresses in Access Lists: NAT used for Source and Destination Addresses



See the following commands for this example:

```
hostname(config)# access-list INSIDE extended permit ip 10.1.1.0 255.255.255.0 host
10.1.1.56
hostname(config)# access-group INSIDE in interface inside
```

Adding an Extended Access List

This section describes how to add an extended access list, and includes the following sections:

- Extended Access List Overview, page 18-5
- Allowing Broadcast and Multicast Traffic through the Transparent Firewall, page 18-6
- Adding an Extended ACE, page 18-7

Extended Access List Overview

An extended access list is made up of one or more ACEs, in which you can specify the line number to insert the ACE, source and destination addresses, and, depending on the ACE type, the protocol, the ports (for TCP or UDP), or the ICMP type (for ICMP). You can identify all of these parameters within the **access-list** command, or you can use object groups for each parameter. This section describes how to identify the parameters within the command. To use object groups, see the "Simplifying Access Lists with Object Grouping" section on page 18-12.

For information about logging options that you can add to the end of the ACE, see the "Logging Access List Activity" section on page 18-21. For information about time range options, see "Scheduling Extended Access List Activation" section on page 18-19.

For TCP and UDP connections for both routed and transparent mode, you do not need an access list to allow returning traffic, because the security appliance allows all returning traffic for established, bidirectional connections. For connectionless protocols such as ICMP, however, the security appliance establishes unidirectional sessions, so you either need access lists to allow ICMP in both directions (by applying access lists to the source and destination interfaces), or you need to enable the ICMP inspection engine. The ICMP inspection engine treats ICMP sessions as bidirectional connections.

You can apply only one access list of each type (extended and EtherType) to each direction of an interface. You can apply the same access lists on multiple interfaces. See Chapter 20, "Permitting or Denying Network Access," for more information about applying an access list to an interface.

Note If you change the access list configuration, and you do not want to wait for existing connections to time out before the new access list information is used, you can clear the connections using the **clear local-host** command.

Allowing Broadcast and Multicast Traffic through the Transparent Firewall

In routed firewall mode, broadcast and multicast traffic is blocked even if you allow it in an access list, including unsupported dynamic routing protocols and DHCP (unless you configure DHCP relay). Transparent firewall mode can allow any IP traffic through. This feature is especially useful in multiple context mode, which does not allow dynamic routing, for example.

Note Because these special types of traffic are connectionless, you need to apply an extended access list to both interfaces, so returning traffic is allowed through.

Table 18-2 lists common traffic types that you can allow through the transparent firewall.

Table 18-2 Transparent Firewall Special Traffic

Traffic Type	Protocol or Port	Notes
DHCP	UDP ports 67 and 68	If you enable the DHCP server, then the security appliance does not pass DHCP packets.
EIGRP	Protocol 88	—
OSPF	Protocol 89	—
Multicast streams	The UDP ports vary depending on the application.	Multicast streams are always destined to a Class D address (224.0.0.0 to 239.x.x.x).
RIP (v1 or v2)	UDP port 520	—

Adding an Extended ACE

When you enter the **access-list** command for a given access list name, the ACE is added to the end of the access list unless you specify the **line** number.

To add an ACE, enter the following command:

```
hostname(config)# access-list access_list_name [line line_number] [extended]
{deny | permit} protocol source_address mask [operator port] dest_address mask
[operator port | icmp_type] [inactive]
```

Tip Enter the access list name in upper case letters so the name is easy to see in the configuration. You might want to name the access list for the interface (for example, INSIDE), or for the purpose for which it is created (for example, NO_NAT or VPN).

Typically, you identify the **ip** keyword for the protocol, but other protocols are accepted. For a list of protocol names, see the "Protocols and Applications" section on page C-11.

Enter the **host** keyword before the IP address to specify a single address. In this case, do not enter a mask. Enter the **any** keyword instead of the address and mask to specify any address.

You can specify the source and destination ports only for the **tcp** or **udp** protocols. For a list of permitted keywords and well-known port assignments, see the "TCP and UDP Ports" section on page C-11. DNS, Discard, Echo, Ident, NTP, RPC, SUNRPC, and Talk each require one definition for TCP and one for UDP. TACACS+ requires one definition for port 49 on TCP.

Use an *operator* to match port numbers used by the source or destination. The permitted operators are as follows:

- **lt**—less than
- **gt**—greater than
- **eq**—equal to
- **neq**—not equal to
- **range**—an inclusive range of values. When you use this operator, specify two port numbers, for example:

  ```
  range 100 200
  ```

You can specify the ICMP type only for the **icmp** protocol. Because ICMP is a connectionless protocol, you either need access lists to allow ICMP in both directions (by applying access lists to the source and destination interfaces), or you need to enable the ICMP inspection engine (see the "Adding an ICMP Type Object Group" section on page 18-16). The ICMP inspection engine treats ICMP sessions as stateful connections. To control ping, specify **echo-reply** (**0**) (security appliance to host) or **echo** (**8**) (host to security appliance). See the "Adding an ICMP Type Object Group" section on page 18-16 for a list of ICMP types.

When you specify a network mask, the method is different from the Cisco IOS software **access-list** command. The security appliance uses a network mask (for example, 255.255.255.0 for a Class C mask). The Cisco IOS mask uses wildcard bits (for example, 0.0.0.255).

To make an ACE inactive, use the **inactive** keyword. To reenable it, enter the entire ACE without the **inactive** keyword. This feature lets you keep a record of an inactive ACE in your configuration to make reenabling easier.

To remove an ACE, enter the **no access-list** command with the entire command syntax string as it appears in the configuration:

```
hostname(config)# no access-list access_list_name [line line_number] [extended]
{deny | permit} protocol source_address mask [operator port] dest_address mask
[operator port | icmp_type] [inactive]
```

If the entry that you are removing is the only entry in the access list, the entire access list is removed.

See the following examples:

The following access list allows all hosts (on the interface to which you apply the access list) to go through the security appliance:

```
hostname(config)# access-list ACL_IN extended permit ip any any
```

The following sample access list prevents hosts on 192.168.1.0/24 from accessing the 209.165.201.0/27 network. All other addresses are permitted.

```
hostname(config)# access-list ACL_IN extended deny tcp 192.168.1.0 255.255.255.0
209.165.201.0 255.255.255.224
hostname(config)# access-list ACL_IN extended permit ip any any
```

If you want to restrict access to only some hosts, then enter a limited permit ACE. By default, all other traffic is denied unless explicitly permitted.

```
hostname(config)# access-list ACL_IN extended permit ip 192.168.1.0 255.255.255.0
209.165.201.0 255.255.255.224
```

The following access list restricts all hosts (on the interface to which you apply the access list) from accessing a website at address 209.165.201.29. All other traffic is allowed.

```
hostname(config)# access-list ACL_IN extended deny tcp any host 209.165.201.29 eq www
hostname(config)# access-list ACL_IN extended permit ip any any
```

Adding an EtherType Access List

Transparent firewall mode only

This section describes how to add an EtherType access list, and includes the following sections:

- EtherType Access List Overview, page 18-8
- Adding an EtherType ACE, page 18-10

EtherType Access List Overview

An EtherType access list is made up of one or more ACEs that specify an EtherType. This section includes the following topics:

- Supported EtherTypes, page 18-9
- Implicit Permit of IP and ARPs Only, page 18-9
- Implicit and Explicit Deny ACE at the End of an Access List, page 18-9
- IPv6 Unsupported, page 18-9
- Using Extended and EtherType Access Lists on the Same Interface, page 18-9
- Allowing MPLS, page 18-10

Supported EtherTypes

An EtherType ACE controls any EtherType identified by a 16-bit hexadecimal number.

EtherType access lists support Ethernet V2 frames.

802.3-formatted frames are not handled by the access list because they use a length field as opposed to a type field.

BPDUs, which are handled by the access list, are the only exception: they are SNAP-encapsulated, and the security appliance is designed to specifically handle BPDUs.

The security appliance receives trunk port (Cisco proprietary) BPDUs. Trunk BPDUs have VLAN information inside the payload, so the security appliance modifies the payload with the outgoing VLAN if you allow BPDUs.



Note If you use failover, you must allow BPDUs on both interfaces with an EtherType access list to avoid bridging loops.

Implicit Permit of IP and ARPs Only

IPv4 traffic is allowed through the transparent firewall automatically from a higher security interface to a lower security interface, without an access list. ARPs are allowed through the transparent firewall in both directions without an access list. ARP traffic can be controlled by ARP inspection.

However, to allow any traffic with EtherTypes other than IPv4 and ARP, you need to apply an EtherType access list, even from a high security to a low security interface.

Because EtherTypes are connectionless, you need to apply the access list to both interfaces if you want traffic to pass in both directions.

Implicit and Explicit Deny ACE at the End of an Access List

For EtherType access lists, the implicit deny at the end of the access list does not affect IP traffic or ARPs; for example, if you allow EtherType 8037, the implicit deny at the end of the access list does not now block any IP traffic that you previously allowed with an extended access list (or implicitly allowed from a high security interface to a low security interface). However, if you *explicitly* deny all traffic with an EtherType ACE, then IP and ARP traffic is denied.

IPv6 Unsupported

EtherType ACEs do not allow IPv6 traffic, even if you specify the IPv6 EtherType.

Using Extended and EtherType Access Lists on the Same Interface

You can apply only one access list of each type (extended and EtherType) to each direction of an interface. You can also apply the same access lists on multiple interfaces.

Allowing MPLS

If you allow MPLS, ensure that Label Distribution Protocol and Tag Distribution Protocol TCP connections are established through the security appliance by configuring both MPLS routers connected to the security appliance to use the IP address on the security appliance interface as the router-id for LDP or TDP sessions. (LDP and TDP allow MPLS routers to negotiate the labels (addresses) used to forward packets.)

On Cisco IOS routers, enter the appropriate command for your protocol, LDP or TDP. The *interface* is the interface connected to the security appliance.

```
hostname(config)# mpls ldp router-id interface force
```

Or

```
hostname(config)# tag-switching tdp router-id interface force
```

Adding an EtherType ACE

To add an EtherType ACE, enter the following command:

```
hostname(config)# access-list access_list_name ethertype {permit | deny} {ipx | bpdu |
mpls-unicast | mpls-multicast | any | hex_number}
```

To remove an EtherType ACE, enter the **no access-list** command with the entire command syntax string as it appears in the configuration:

```
ehostname(config)# no access-list access_list_name ethertype {permit | deny} {ipx | bpdu |
mpls-unicast | mpls-multicast | any | hex_number}
```

The *hex_number* is any EtherType that can be identified by a 16-bit hexadecimal number greater than or equal to 0x600. See RFC 1700, "Assigned Numbers," at http://www.ietf.org/rfc/rfc1700.txt for a list of EtherTypes.

Note If an EtherType access list is configured to **deny all**, all ethernet frames are discarded. Only physical protocol traffic, such as auto-negotiation, is still allowed.

When you enter the **access-list** command for a given access list name, the ACE is added to the end of the access list.

Tip Enter the *access_list_name* in upper case letters so the name is easy to see in the configuration. You might want to name the access list for the interface (for example, INSIDE), or for the purpose (for example, MPLS or IPX).

For example, the following sample access list allows common EtherTypes originating on the inside interface:

```
hostname(config)# access-list ETHER ethertype permit ipx
hostname(config)# access-list ETHER ethertype permit bpdu
hostname(config)# access-list ETHER ethertype permit mpls-unicast
hostname(config)# access-group ETHER in interface inside
```

The following access list allows some EtherTypes through the security appliance, but denies IPX:

```
hostname(config)# access-list ETHER ethertype deny ipx
```

```
hostname(config)# access-list ETHER ethertype permit 0x1234
hostname(config)# access-list ETHER ethertype permit bpdu
hostname(config)# access-list ETHER ethertype permit mpls-unicast
hostname(config)# access-group ETHER in interface inside
hostname(config)# access-group ETHER in interface outside
```

The following access list denies traffic with EtherType 0x1256, but allows all others on both interfaces:

```
hostname(config)# access-list nonIP ethertype deny 1256
hostname(config)# access-list nonIP ethertype permit any
hostname(config)# access-group ETHER in interface inside
hostname(config)# access-group ETHER in interface outside
```

Adding a Standard Access List

Single context mode only

Standard access lists identify the destination IP addresses of OSPF routes, and can be used in a route map for OSPF redistribution. Standard access lists cannot be applied to interfaces to control traffic.

The following command adds a standard ACE. To add another ACE at the end of the access list, enter another **access-list** command specifying the same access list name. Apply the access list using the "Defining Route Maps" section on page 10-7.

To add an ACE, enter the following command:

```
hostname(config)# access-list access_list_name standard {deny | permit} {any | ip_address
mask}
```

The following sample access list identifies routes to 192.168.1.0/24:

```
hostname(config)# access-list OSPF standard permit 192.168.1.0 255.255.255.0
```

To remove an ACE, enter the **no access-list** command with the entire command syntax string as it appears in the configuration:

```
hostname(config)# no access-list access_list_name standard {deny | permit} {any |
ip_address mask}
```

Adding a Webtype Access List

Webtype access lists are access lists that are added to a configuration that supports filtering for clientless SSL VPN.

You can use the following wildcard characters to define more than one wildcard in the Webtype access list entry:

- Enter an asterisk "*" to match no characters or any number of characters.
- Enter a question mark "?" to match any one character exactly.
- Enter square brackets "[]" to create a range operator that matches any one character in a range.

To add an access list to the configuration that supports filtering for WebVPN, enter the following command:

```
hostname(config)# access-list access_list_name webtype {deny | permit} url [url_string | any]
```

To remove a Webtype access list, enter the **no access-list** command with the entire syntax string as it appears in the configuration:

```
hostname(config)# access-list access_list_name webtype {deny | permit} url [url_string | any]
```

The following scenario shows how to enforce a webtype access list to disable access to specific CIFS shares.

In this scenario we have a root folder named "shares" that contains two sub-folders named "Marketing_Reports" and "Sales_Reports." We want to specifically deny access to the "shares/Marketing_Reports" folder.

```
access-list CIFS_Avoid webtype deny url cifs://172.16.10.40/shares/Marketing_Reports.
```

However, due to the implicit "deny all," the above access list makes all of the sub-folders inaccessible ("shares/Sales_Reports" and "shares/Marketing_Reports"), including the root folder ("shares").

To fix the problem, add a new access list to allow access to the root folder and the remaining sub-folders.

```
access-list CIFS_Allow webtype permit url cifs://172.16.10.40/shares*
```

For information about logging options that you can add to the end of the ACE, see the "Logging Access List Activity" section on page 18-21.

Examples

The examples in this section show how to use wildcards in Webtype access lists.

- The following example matches URLs such as http://www.cisco.com/ and http://wwz.caco.com/:

  ```
  access-list test webtype permit url http://ww?.c*co*/
  ```

- The following example matches URLs such as http://www.cisco.com and ftp://wwz.carrier.com:

  ```
  access-list test webtype permit url *://ww?.c*co*/
  ```

- The following example matches URLs such as http://www.cisco.com:80 and https://www.cisco.com:81:

  ```
  access-list test webtype permit url *://ww?.c*co*:8[01]/
  ```

  The range operator "[]" in the preceding example specifies that either character **0** or **1** can occur.

- The following example matches URLs such as http://www.google.com and http://www.boogie.com:

  ```
  access-list test webtype permit url http://www.[a-zloo?*/
  ```

  The range operator "[]" in the preceding example specifies that any character in the range from **a** to **z** can occur.

- The following example matches URLs such as http://www.cisco.com/anything/crazy/url/ddtscgiz:

  ```
  access-list test webtype permit url htt*://*/*cgi?*
  ```

Note To match any http URL, you must enter **http://*/*** instead of the former method of entering http://*.

Simplifying Access Lists with Object Grouping

This section describes how to use object grouping to simplify access list creation and maintenance.

This section includes the following topics:

- How Object Grouping Works, page 18-13
- Adding Object Groups, page 18-13
- Nesting Object Groups, page 18-16
- Displaying Object Groups, page 18-18
- Removing Object Groups, page 18-19
- Using Object Groups with an Access List, page 18-17

How Object Grouping Works

By grouping like-objects together, you can use the object group in an ACE instead of having to enter an ACE for each object separately. You can create the following types of object groups:

- Protocol
- Network
- Service
- ICMP type

For example, consider the following three object groups:

- MyServices—Includes the TCP and UDP port numbers of the service requests that are allowed access to the internal network
- TrustedHosts—Includes the host and network addresses allowed access to the greatest range of services and servers
- PublicServers—Includes the host addresses of servers to which the greatest access is provided

After creating these groups, you could use a single ACE to allow trusted hosts to make specific service requests to a group of public servers.

You can also nest object groups in other object groups.

Note The ACE system limit applies to expanded access lists. If you use object groups in ACEs, the number of actual ACEs that you enter is fewer, but the number of expanded ACEs is the same as without object groups. In many cases, object groups create more ACEs than if you added them manually, because creating ACEs manually leads you to summarize addresses more than an object group does. To view the number of expanded ACEs in an access list, enter the **show access-list** *access_list_name* command.

Adding Object Groups

This section describes how to add object groups.

This section includes the following topics:

- Adding a Protocol Object Group, page 18-14
- Adding a Network Object Group, page 18-14
- Adding a Service Object Group, page 18-15
- Adding an ICMP Type Object Group, page 18-16

Adding a Protocol Object Group

To add or change a protocol object group, perform the following steps. After you add the group, you can add more objects as required by following this procedure again for the same group name and specifying additional objects. You do not need to reenter existing objects; the commands you already set remain in place unless you remove them with the **no** form of the command.

To add a protocol group, perform the following steps:

Step 1 To add a protocol group, enter the following command:

```
hostname(config)# object-group protocol grp_id
```

The *grp_id* is a text string up to 64 characters in length.

The prompt changes to protocol configuration mode.

Step 2 (Optional) To add a description, enter the following command:

```
hostname(config-protocol)# description text
```

The description can be up to 200 characters.

Step 3 To define the protocols in the group, enter the following command for each protocol:

```
hostname(config-protocol)# protocol-object protocol
```

The *protocol* is the numeric identifier of the specific IP protocol (1 to 254) or a keyword identifier (for example, **icmp**, **tcp**, or **udp**). To include all IP protocols, use the keyword **ip**. For a list of protocols you can specify, see the "Protocols and Applications" section on page C-11.

For example, to create a protocol group for TCP, UDP, and ICMP, enter the following commands:

```
hostname(config)# object-group protocol tcp_udp_icmp
hostname(config-protocol)# protocol-object tcp
hostname(config-protocol)# protocol-object udp
hostname(config-protocol)# protocol-object icmp
```

Adding a Network Object Group

To add or change a network object group, perform the following steps. After you add the group, you can add more objects as required by following this procedure again for the same group name and specifying additional objects. You do not need to reenter existing objects; the commands you already set remain in place unless you remove them with the **no** form of the command.

Note A network object group supports IPv4 and IPv6 addresses, depending on the type of access list. For more information about IPv6 access lists, see "Configuring IPv6 Access Lists" section on page 13-6.

To add a network group, perform the following steps:

Step 1 To add a network group, enter the following command:

```
hostname(config)# object-group network grp_id
```

The *grp_id* is a text string up to 64 characters in length.

The prompt changes to network configuration mode.

Step 2 (Optional) To add a description, enter the following command:

```
hostname(config-network)# description text
```

The description can be up to 200 characters.

Step 3 To define the networks in the group, enter the following command for each network or address:

```
hostname(config-network)# network-object {host ip_address | ip_address mask}
```

For example, to create network group that includes the IP addresses of three administrators, enter the following commands:

```
hostname(config)# object-group network admins
hostname(config-network)# description Administrator Addresses
hostname(config-network)# network-object host 10.1.1.4
hostname(config-network)# network-object host 10.1.1.78
hostname(config-network)# network-object host 10.1.1.34
```

Adding a Service Object Group

To add or change a service object group, perform the following steps. After you add the group, you can add more objects as required by following this procedure again for the same group name and specifying additional objects. You do not need to reenter existing objects; the commands you already set remain in place unless you remove them with the **no** form of the command.

To add a service group, perform the following steps:

Step 1 To add a service group, enter the following command:

```
hostname(config)# object-group service grp_id {tcp | udp | tcp-udp}
```

The *grp_id* is a text string up to 64 characters in length.

Specify the protocol for the services (ports) you want to add, either **tcp**, **udp**, or **tcp-udp** keywords. Enter **tcp-udp** keyword if your service uses both TCP and UDP with the same port number, for example, DNS (port 53).

The prompt changes to service configuration mode.

Step 2 (Optional) To add a description, enter the following command:

```
hostname(config-service)# description text
```

The description can be up to 200 characters.

Step 3 To define the ports in the group, enter the following command for each port or range of ports:

```
hostname(config-service)# port-object {eq port | range begin_port end_port}
```

For a list of permitted keywords and well-known port assignments, see the "Protocols and Applications" section on page C-11.

For example, to create service groups that include DNS (TCP/UDP), LDAP (TCP), and RADIUS (UDP), enter the following commands:

```
hostname(config)# object-group service services1 tcp-udp
hostname(config-service)# description DNS Group
hostname(config-service)# port-object eq domain
```

```
hostname(config-service)# object-group service services2 udp
hostname(config-service)# description RADIUS Group
hostname(config-service)# port-object eq radius
hostname(config-service)# port-object eq radius-acct

hostname(config-service)# object-group service services3 tcp
hostname(config-service)# description LDAP Group
hostname(config-service)# port-object eq ldap
```

Adding an ICMP Type Object Group

To add or change an ICMP type object group, perform the following steps. After you add the group, you can add more objects as required by following this procedure again for the same group name and specifying additional objects. You do not need to reenter existing objects; the commands you already set remain in place unless you remove them with the **no** form of the command.

To add an ICMP type group, perform the following steps:

Step 1 To add an ICMP type group, enter the following command:

```
hostname(config)# object-group icmp-type grp_id
```

The *grp_id* is a text string up to 64 characters in length.

The prompt changes to ICMP type configuration mode.

Step 2 (Optional) To add a description, enter the following command:

```
hostname(config-icmp-type)# description text
```

The description can be up to 200 characters.

Step 3 To define the ICMP types in the group, enter the following command for each type:

```
hostname(config-icmp-type)# icmp-object icmp_type
```

See the "ICMP Types" section on page C-15 for a list of ICMP types.

For example, to create an ICMP type group that includes echo-reply and echo (for controlling ping), enter the following commands:

```
hostname(config)# object-group icmp-type ping
hostname(config-service)# description Ping Group
hostname(config-icmp-type)# icmp-object echo
hostname(config-icmp-type)# icmp-object echo-reply
```

Nesting Object Groups

To nest an object group within another object group of the same type, first create the group that you want to nest according to the "Adding Object Groups" section on page 18-13. Then perform the following steps:

Step 1 To add or edit an object group under which you want to nest another object group, enter the following command:

```
hostname(config)# object-group {{protocol | network | icmp-type} grp_id | service grp_id
{tcp | udp | tcp-udp}}
```

Step 2 To add the specified group under the object group you specified in Step 1, enter the following command:

```
hostname(config-group_type)# group-object grp_id
```

The nested group must be of the same type.

You can mix and match nested group objects and regular objects within an object group.

For example, you create network object groups for privileged users from various departments:

```
hostname(config)# object-group network eng
hostname(config-network)# network-object host 10.1.1.5
hostname(config-network)# network-object host 10.1.1.9
hostname(config-network)# network-object host 10.1.1.89

hostname(config-network)# object-group network hr
hostname(config-network)# network-object host 10.1.2.8
hostname(config-network)# network-object host 10.1.2.12

hostname(config-network)# object-group network finance
hostname(config-network)# network-object host 10.1.4.89
hostname(config-network)# network-object host 10.1.4.100
```

You then nest all three groups together as follows:

```
hostname(config)# object-group network admin
hostname(config-network)# group-object eng
hostname(config-network)# group-object hr
hostname(config-network)# group-object finance
```

You only need to specify the admin object group in your ACE as follows:

```
hostname(config)# access-list ACL_IN extended permit ip object-group admin host
209.165.201.29
```

Using Object Groups with an Access List

To use object groups in an access list, replace the normal protocol (*protocol*), network (*source_address mask*, etc.), service (*operator port*), or ICMP type (*icmp_type*) parameter with **object-group** *grp_id* parameter.

For example, to use object groups for all available parameters in the **access-list** {**tcp** | **udp**} command, enter the following command:

```
hostname(config)# access-list access_list_name [line line_number] [extended] {deny |
permit} {tcp | udp} object-group nw_grp_id [object-group svc_grp_id] object-group
nw_grp_id [object-group svc_grp_id] [log [[level] [interval secs] | disable | default]]
[inactive | time-range time_range_name]
```

You do not have to use object groups for all parameters; for example, you can use an object group for the source address, but identify the destination address with an address and mask.

The following normal access list that does not use object groups restricts several hosts on the inside network from accessing several web servers. All other traffic is allowed.

```
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.4 host 209.165.201.29
eq www
```

```
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.78 host 209.165.201.29
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.89 host 209.165.201.29
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.4 host 209.165.201.16
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.78 host 209.165.201.16
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.89 host 209.165.201.16
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.4 host 209.165.201.78
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.78 host 209.165.201.78
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.89 host 209.165.201.78
eq www
hostname(config)# access-list ACL_IN extended permit ip any any
hostname(config)# access-group ACL_IN in interface inside
```

If you make two network object groups, one for the inside hosts, and one for the web servers, then the configuration can be simplified and can be easily modified to add more hosts:

```
hostname(config)# object-group network denied
hostname(config-network)# network-object host 10.1.1.4
hostname(config-network)# network-object host 10.1.1.78
hostname(config-network)# network-object host 10.1.1.89

hostname(config-network)# object-group network web
hostname(config-network)# network-object host 209.165.201.29
hostname(config-network)# network-object host 209.165.201.16
hostname(config-network)# network-object host 209.165.201.78

hostname(config-network)# access-list ACL_IN extended deny tcp object-group denied
object-group web eq www
hostname(config)# access-list ACL_IN extended permit ip any any
hostname(config)# access-group ACL_IN in interface inside
```

Displaying Object Groups

To display a list of the currently configured object groups, enter the following command:

```
hostname(config)# show object-group [protocol | network | service | icmp-type | id grp_id]
```

If you enter the command without any parameters, the system displays all configured object groups.

The following is sample output from the **show object-group** command:

```
hostname# show object-group
object-group network ftp_servers
  description: This is a group of FTP servers
  network-object host 209.165.201.3
  network-object host 209.165.201.4
object-group network TrustedHosts
  network-object host 209.165.201.1
  network-object 192.168.1.0 255.255.255.0
  group-object ftp_servers
```

Removing Object Groups

To remove an object group, enter one of the following commands.

Note You cannot remove an object group or make an object group empty if it is used in an access list.

- To remove a specific object group, enter the following command:

  ```
  hostname(config)# no object-group grp_id
  ```

- To remove all object groups of the specified type, enter the following command:

  ```
  hostname(config)# clear object-group [protocol | network | services | icmp-type]
  ```

  If you do not enter a type, all object groups are removed.

Adding Remarks to Access Lists

You can include remarks about entries in any access list, including extended, EtherType, and standard access lists. The remarks make the access list easier to understand.

To add a remark after the last **access-list** command you entered, enter the following command:

```
hostname(config)# access-list access_list_name remark text
```

If you enter the remark before any **access-list** command, then the remark is the first line in the access list.

If you delete an access list using the **no access-list** *access_list_name* command, then all the remarks are also removed.

The text can be up to 100 characters in length. You can enter leading spaces at the beginning of the text. Trailing spaces are ignored.

For example, you can add remarks before each ACE, and the remark appears in the access list in this location. Entering a dash (-) at the beginning of the remark helps set it apart from ACEs.

```
hostname(config)# access-list OUT remark - this is the inside admin address
hostname(config)# access-list OUT extended permit ip host 209.168.200.3 any
hostname(config)# access-list OUT remark - this is the hr admin address
hostname(config)# access-list OUT extended permit ip host 209.168.200.4 any
```

Scheduling Extended Access List Activation

You can schedule each ACE to be activated at specific times of the day and week by applying a time range to the ACE. This section includes the following topics:

- Adding a Time Range, page 18-19
- Applying the Time Range to an ACE, page 18-20

Adding a Time Range

To add a time range to implement a time-based access list, perform the following steps:

Step 1 Identify the time-range name by entering the following command:

```
hostname(config)# time-range name
```

Step 2 Specify the time range as either a recurring time range or an absolute time range.

Note Users could experience a delay of approximately 80 to 100 seconds after the specified end time for the ACL to become inactive. For example, if the specified end time is 3:50, because the end time is inclusive, the command is picked up anywhere between 3:51:00 and 3:51:59. After the command is picked up, the security appliance finishes any currently running task and then services the command to deactivate the ACL.

Multiple periodic entries are allowed per **time-range** command. If a **time-range** command has both **absolute** and **periodic** values specified, then the **periodic** commands are evaluated only after the **absolute** start time is reached, and are not further evaluated after the **absolute** end time is reached.

- Recurring time range:

  ```
  hostname(config-time-range)# periodic days-of-the-week time to [days-of-the-week] time
  ```

  You can specify the following values for *days-of-the-week*:

  - **monday**, **tuesday**, **wednesday**, **thursday**, **friday**, **saturday**, and **sunday**.
  - **daily**
  - **weekdays**
  - **weekend**

  The *time* is in the format *hh*:*mm*. For example, 8:00 is 8:00 a.m. and 20:00 is 8:00 p.m.

- Absolute time range:

  ```
  hostname(config-time-range)# absolute start time date [end time date]
  ```

  The *time* is in the format *hh*:*mm*. For example, 8:00 is 8:00 a.m. and 20:00 is 8:00 p.m.

  The *date* is in the format *day month year*; for example, **1 january 2006**.

The following is an example of an absolute time range beginning at 8:00 a.m. on January 1, 2006. Because no end time and date are specified, the time range is in effect indefinitely.

```
hostname(config)# time-range for2006
hostname(config-time-range)# absolute start 8:00 1 january 2006
```

The following is an example of a weekly periodic time range from 8:00 a.m. to 6:00 p.m on weekdays.:

```
hostname(config)# time-range workinghours
hostname(config-time-range)# periodic weekdays 8:00 to 18:00
```

Applying the Time Range to an ACE

To apply the time range to an ACE, use the following command:

```
hostname(config)# access-list access_list_name [extended] {deny | permit}...[time-range
name]
```

See the "Adding an Extended Access List" section on page 18-5 for complete **access-list** command syntax.

Note If you also enable logging for the ACE, use the **log** keyword before the **time-range** keyword. If you disable the ACE using the **inactive** keyword, use the **inactive** keyword as the last keyword.

The following example binds an access list named "Sales" to a time range named "New_York_Minute."

```
hostname(config)# access-list Sales line 1 extended deny tcp host 209.165.200.225 host
209.165.201.1 time-range New_York_Minute
```

Logging Access List Activity

This section describes how to configure access list logging for extended access lists and Webtype access lists.

This section includes the following topics:

- Access List Logging Overview, page 18-21
- Configuring Logging for an Access Control Entry, page 18-22
- Managing Deny Flows, page 18-23

Access List Logging Overview

By default, when traffic is denied by an extended ACE or a Webtype ACE, the security appliance generates system message 106023 for each denied packet, in the following form:

```
%ASA|PIX-4-106023: Deny protocol src [interface_name:source_address/source_port] dst
interface_name:dest_address/dest_port [type {string}, code {code}] by access_group acl_id
```

If the security appliance is attacked, the number of system messages for denied packets can be very large. We recommend that you instead enable logging using system message 106100, which provides statistics for each ACE and lets you limit the number of system messages produced. Alternatively, you can disable all logging.

Note Only ACEs in the access list generate logging messages; the implicit deny at the end of the access list does not generate a message. If you want all denied traffic to generate messages, add the implicit ACE manually to the end of the access list, as follows.

```
hostname(config)# access-list TEST deny ip any any log
```

The **log** options at the end of the extended **access-list** command lets you to set the following behavior:

- Enable message 106100 instead of message 106023
- Disable all logging
- Return to the default logging using message 106023

System message 106100 is in the following form:

```
%ASA|PIX-n-106100: access-list acl_id {permitted | denied} protocol
interface_name/source_address(source_port) -> interface_name/dest_address(dest_port)
hit-cnt number ({first hit | number-second interval})
```

When you enable logging for message 106100, if a packet matches an ACE, the security appliance creates a flow entry to track the number of packets received within a specific interval. The security appliance generates a system message at the first hit and at the end of each interval, identifying the total number of hits during the interval. At the end of each interval, the security appliance resets the hit count to 0. If no packets match the ACE during an interval, the security appliance deletes the flow entry.

A flow is defined by the source and destination IP addresses, protocols, and ports. Because the source port might differ for a new connection between the same two hosts, you might not see the same flow increment because a new flow was created for the connection. See the "Managing Deny Flows" section on page 18-23 to limit the number of logging flows.

Permitted packets that belong to established connections do not need to be checked against access lists; only the initial packet is logged and included in the hit count. For connectionless protocols, such as ICMP, all packets are logged even if they are permitted, and all denied packets are logged.

See the *Cisco Security Appliance Logging Configuration and System Log Messages* for detailed information about this system message.

Configuring Logging for an Access Control Entry

To configure logging for an ACE, see the following information about the **log** option:

```
hostname(config)# access-list access_list_name [extended] {deny | permit}...[log [[level]
[interval secs] | disable | default]]
```

See the "Adding an Extended Access List" section on page 18-5 and "Adding a Webtype Access List" section on page 18-11 for complete **access-list** command syntax.

If you enter the **log** option without any arguments, you enable system log message 106100 at the default level (6) and for the default interval (300 seconds). See the following options:

- *level*—A severity level between 0 and 7. The default is 6.
- **interval** *secs*—The time interval in seconds between system messages, from 1 to 600. The default is 300. This value is also used as the timeout value for deleting an inactive flow.
- **disable**—Disables all access list logging.
- **default**—Enables logging to message 106023. This setting is the same as having no **log** option.

For example, you configure the following access list:

```
hostname(config)# access-list outside-acl permit ip host 1.1.1.1 any log 7 interval 600
hostname(config)# access-list outside-acl permit ip host 2.2.2.2 any
hostname(config)# access-list outside-acl deny ip any any log 2
hostname(config)# access-group outside-acl in interface outside
```

When a packet is permitted by the first ACE of outside-acl, the security appliance generates the following system message:

```
%ASA|PIX-7-106100: access-list outside-acl permitted tcp outside/1.1.1.1(12345) ->
inside/192.168.1.1(1357) hit-cnt 1 (first hit)
```

Although 20 additional packets for this connection arrive on the outside interface, the traffic does not have to be checked against the access list, and the hit count does not increase.

If one more connection by the same host is initiated within the specified 10 minute interval (and the source and destination ports remain the same), then the hit count is incremented by 1 and the following message is displayed at the end of the 10 minute interval:

```
%ASA|PIX-7-106100: access-list outside-acl permitted tcp outside/1.1.1.1(12345)->
inside/192.168.1.1(1357) hit-cnt 2 (600-second interval)
```

When a packet is denied by the third ACE, the security appliance generates the following system message:

```
%ASA|PIX-2-106100: access-list outside-acl denied ip outside/3.3.3.3(12345) ->
inside/192.168.1.1(1357) hit-cnt 1 (first hit)
```

20 additional attempts within a 5 minute interval (the default) result in the following message at the end of 5 minutes:

```
%ASA|PIX-2-106100: access-list outside-acl denied ip outside/3.3.3.3(12345) ->
inside/192.168.1.1(1357) hit-cnt 21 (300-second interval)
```

Managing Deny Flows

When you enable logging for message 106100, if a packet matches an ACE, the security appliance creates a flow entry to track the number of packets received within a specific interval. The security appliance has a maximum of 32 K logging flows for ACEs. A large number of flows can exist concurrently at any point of time. To prevent unlimited consumption of memory and CPU resources, the security appliance places a limit on the number of concurrent *deny* flows; the limit is placed only on deny flows (and not permit flows) because they can indicate an attack. When the limit is reached, the security appliance does not create a new deny flow for logging until the existing flows expire.

For example, if someone initiates a DoS attack, the security appliance can create a large number of deny flows in a short period of time. Restricting the number of deny flows prevents unlimited consumption of memory and CPU resources.

When you reach the maximum number of deny flows, the security appliance issues system message 106100:

```
%ASA|PIX-1-106101: The number of ACL log deny-flows has reached limit (number).
```

To configure the maximum number of deny flows and to set the interval between deny flow alert messages (106101), enter the following commands:

- To set the maximum number of deny flows permitted per context before the security appliance stops logging, enter the following command:

  ```
  hostname(config)# access-list deny-flow-max number
  ```

  The *number* is between 1 and 4096. 4096 is the default.

- To set the amount of time between system messages (number 106101) that identify that the maximum number of deny flows was reached, enter the following command:

  ```
  hostname(config)# access-list alert-interval secs
  ```

  The *seconds* are between 1 and 3600, and 300 is the default.



CHAPTER 19

Configuring NAT

This chapter describes Network Address Translation, and includes the following sections:

- NAT Overview, page 19-1
- Configuring NAT Control, page 19-18
- Using Dynamic NAT and PAT, page 19-19
- Using Static NAT, page 19-28
- Using Static PAT, page 19-29
- Bypassing NAT, page 19-32
- NAT Examples, page 19-36

NAT Overview

This section describes how NAT works on the security appliance, and includes the following topics:

- Introduction to NAT, page 19-1
- NAT Control, page 19-5
- NAT Types, page 19-6
- Policy NAT, page 19-11
- NAT and Same Security Level Interfaces, page 19-15
- Order of NAT Commands Used to Match Real Addresses, page 19-16
- Mapped Address Guidelines, page 19-16
- DNS and NAT, page 19-17

Introduction to NAT

Address translation substitutes the real address in a packet with a mapped address that is routable on the destination network. NAT is composed of two steps: the process by which a real address is translated into a mapped address, and the process to undo translation for returning traffic.

The security appliance translates an address when a NAT rule matches the traffic. If no NAT rule matches, processing for the packet continues. The exception is when you enable NAT control. NAT control requires that packets traversing from a higher security interface (inside) to a lower security

interface (outside) match a NAT rule, or processing for the packet stops. See the "Security Level Overview" section on page 8-1 for more information about security levels. See the "NAT Control" section on page 19-5 for more information about NAT control.

Note In this document, all types of translation are referred to as NAT. When describing NAT, the terms *inside* and *outside* represent the security relationship between any two interfaces. The higher security level is inside and the lower security level is outside. For example, interface 1 is at 60 and interface 2 is at 50; therefore, interface 1 is "inside" and interface 2 is "outside."

Some of the benefits of NAT are as follows:

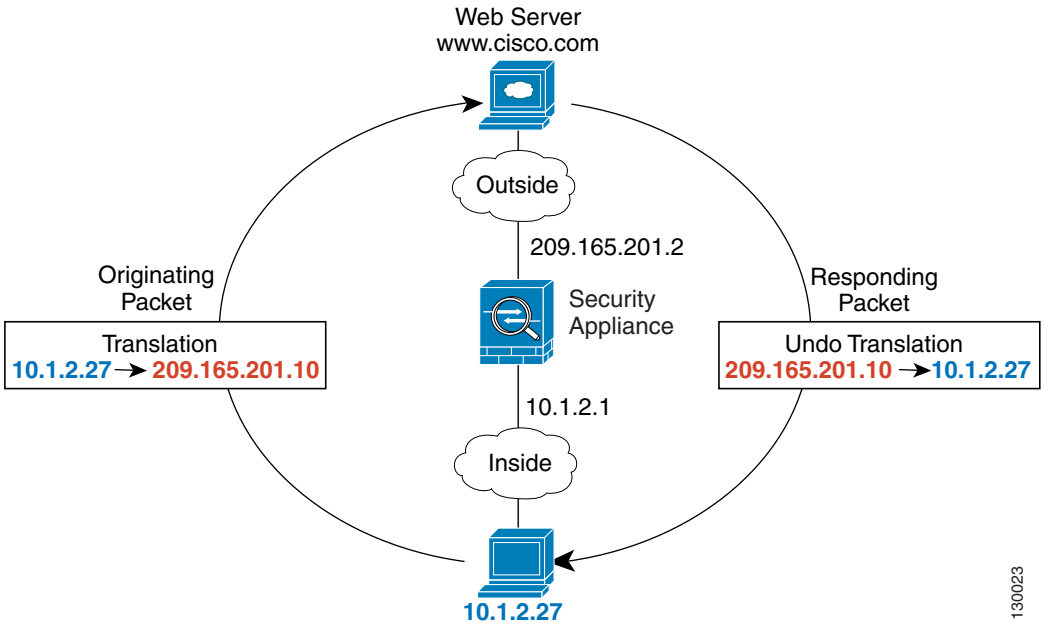
- You can use private addresses on your inside networks. Private addresses are not routable on the Internet. See the "Private Networks" section on page C-2 for more information.
- NAT hides the real addresses from other networks, so attackers cannot learn the real address of a host.
- You can resolve IP routing problems such as overlapping addresses.

See Table 26-1 on page 26-3 for information about protocols that do not support NAT.

NAT in Routed Mode

Figure 19-1 shows a typical NAT example in routed mode, with a private network on the inside. When the inside host at 10.1.2.27 sends a packet to a web server, the real source address, 10.1.2.27, of the packet is changed to a mapped address, 209.165.201.10. When the server responds, it sends the response to the mapped address, 209.165.201.10, and the security appliance receives the packet. The security appliance then changes the translation of the mapped address, 209.165.201.10 back to the real address, 10.1.2.27 before sending it to the host.

Figure 19-1 NAT Example: Routed Mode



See the following commands for this example:

```
hostname(config)# nat (inside) 1 10.1.2.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.1-209.165.201.15
```

NAT in Transparent Mode

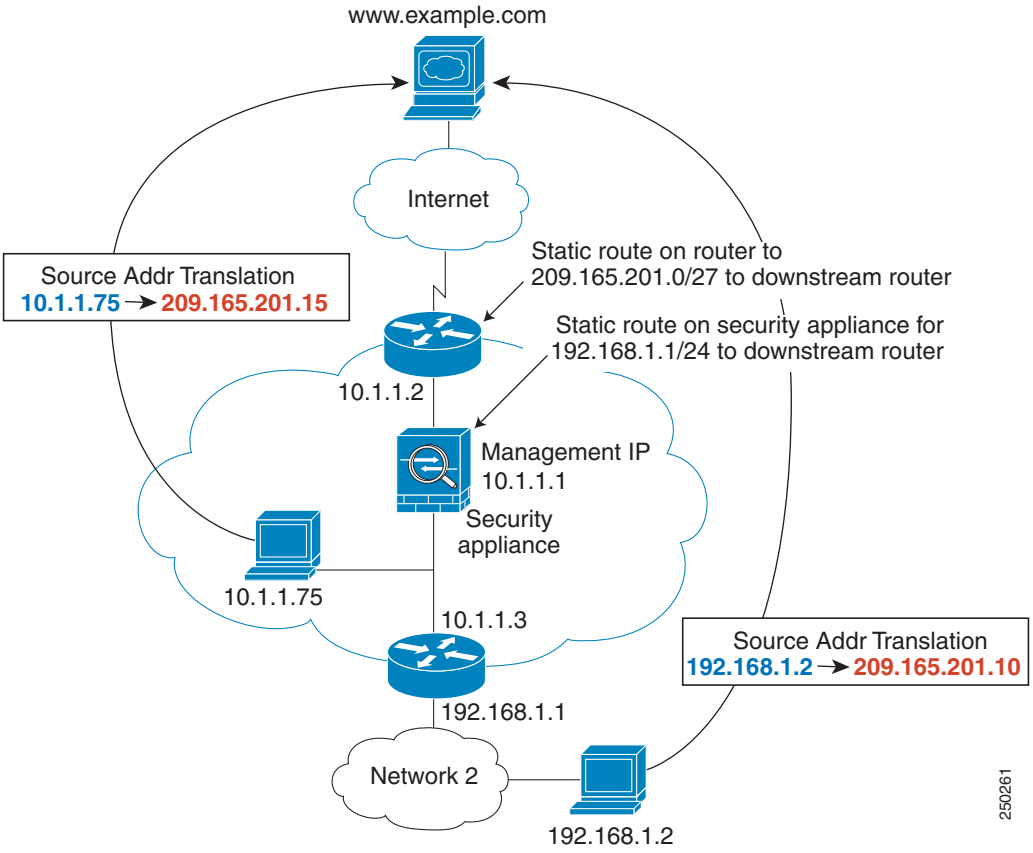
Using NAT in transparent mode eliminates the need for the upstream or downstream routers to perform NAT for their networks. For example, a transparent firewall security appliance is useful between two VRFs so you can establish BGP neighbor relations between the VRFs and the global table. However, NAT per VRF might not be supported. In this case, using NAT in transparent mode is essential.

NAT in transparent mode has the following requirements and limitations:

- When the mapped addresses are not on the same network as the transparent firewall, then on the upstream router, you need to add a static route for the mapped addresses that points to the downstream router (through the security appliance).
- If the real destination address is not directly-connected to the security appliance, then you also need to add a static route on the security appliance for the real destination address that points to the downstream router. Without NAT, traffic from the upstream router to the downstream router does not need any routes on the security appliance because it uses the MAC address table. NAT, however, causes the security appliance to use a route lookup instead of a MAC address lookup, so it needs a static route to the downstream router.
- The **alias** command is not supported.
- Because the transparent firewall does not have any interface IP addresses, you cannot use interface PAT.
- ARP inspection is not supported. Moreover, if for some reason a host on one side of the firewall sends an ARP request to a host on the other side of the firewall, and the initiating host real address is mapped to a different address on the same subnet, then the real address remains visible in the ARP request.

Figure 19-2 shows a typical NAT scenario in transparent mode, with the same network on the inside and outside interfaces. The transparent firewall in this scenario is performing the NAT service so that the upstream router does not have to perform NAT. When the inside host at 10.1.1.27 sends a packet to a web server, the real source address of the packet, 10.1.1.27, is changed to a mapped address, 209.165.201.10. When the server responds, it sends the response to the mapped address, 209.165.201.10, and the security appliance receives the packet because the upstream router includes this mapped network in a static route directed through the security appliance. The security appliance then undoes the translation of the mapped address, 209.165.201.10 back to the real address, 10.1.1.1.27. Because the real address is directly-connected, the security appliance sends it directly to the host. For host 192.168.1.2, the same process occurs, except that the security appliance looks up the route in its route table, and sends the packet to the downstream router at 10.1.1.3 based on the static route.

Figure 19-2 NAT Example: Transparent Mode



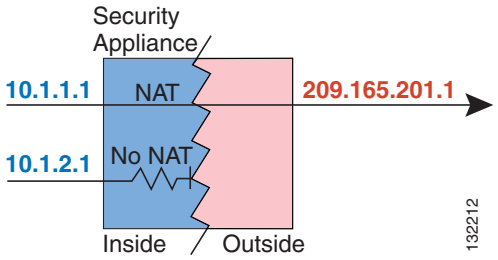
See the following commands for this example:

```
hostname(config)# route inside 192.168.1.0 255.255.255.0 10.1.1.3 1
hostname(config)# nat (inside) 1 10.1.1.0 255.255.255.0
hostname(config)# nat (inside) 1 192.168.1.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.1-209.165.201.15
```

NAT Control

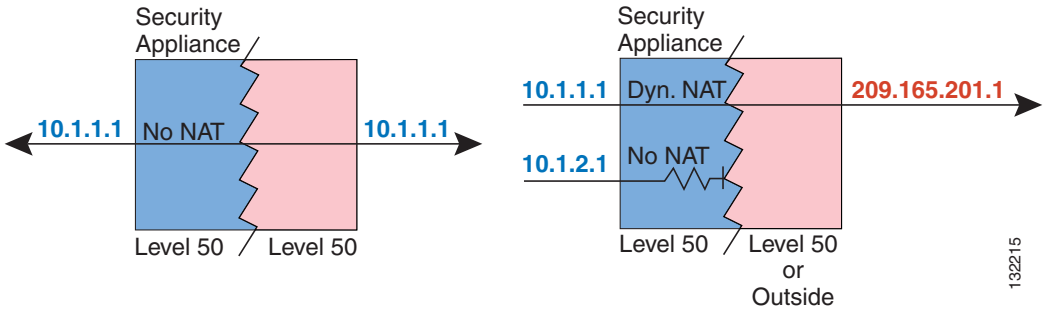
NAT control requires that packets traversing from an inside interface to an outside interface match a NAT rule; for any host on the inside network to access a host on the outside network, you must configure NAT to translate the inside host address, as shown in Figure 19-3.

Figure 19-3 NAT Control and Outbound Traffic



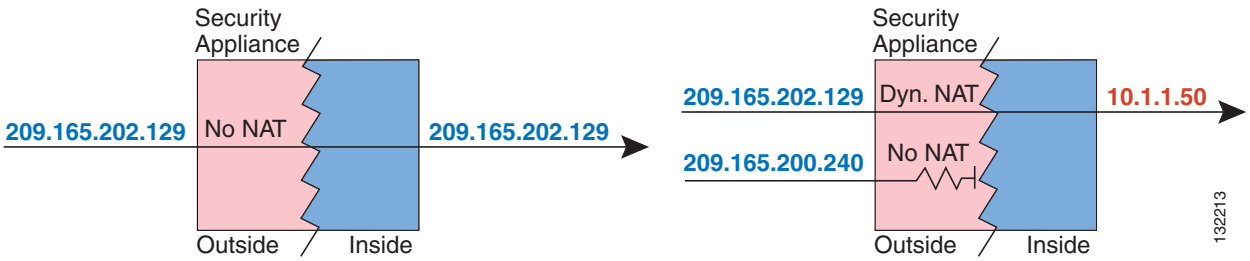
Interfaces at the same security level are not required to use NAT to communicate. However, if you configure dynamic NAT or PAT on a same security interface, then all traffic from the interface to a same security interface or an outside interface must match a NAT rule, as shown in Figure 19-4.

Figure 19-4 NAT Control and Same Security Traffic



Similarly, if you enable outside dynamic NAT or PAT, then all outside traffic must match a NAT rule when it accesses an inside interface (see Figure 19-5).

Figure 19-5 NAT Control and Inbound Traffic



Static NAT does not cause these restrictions.

By default, NAT control is disabled; therefore, you do not need to perform NAT on any networks unless you want to do so. If you upgraded from an earlier version of software, however, NAT control might be enabled on your system. Even with NAT control disabled, you need to perform NAT on any addresses for which you configure dynamic NAT. See the "Dynamic NAT and PAT Implementation" section on page 19-19 for more information about how dynamic NAT is applied.

If you want the added security of NAT control but do not want to translate inside addresses in some cases, you can apply a NAT exemption or identity NAT rule on those addresses. (See the "Bypassing NAT" section on page 19-32 for more information).

To configure NAT control, see the "Configuring NAT Control" section on page 19-18.

Note In multiple context mode, the packet classifier might rely on the NAT configuration to assign packets to contexts if you do not enable unique MAC addresses for shared interfaces. See the "How the Security Appliance Classifies Packets" section on page 4-3 for more information about the relationship between the classifier and NAT.

NAT Types

This section describes the available NAT types, and includes the following topics:

- Dynamic NAT, page 19-6
- PAT, page 19-8
- Static NAT, page 19-9
- Static PAT, page 19-9
- Bypassing NAT When NAT Control is Enabled, page 19-10

You can implement address translation as dynamic NAT, Port Address Translation, static NAT, static PAT, or as a mix of these types. You can also configure rules to bypass NAT; for example, to enable NAT control when you do not want to perform NAT.

Dynamic NAT

Dynamic NAT translates a group of real addresses to a pool of mapped addresses that are routable on the destination network. The mapped pool may include fewer addresses than the real group. When a host you want to translate accesses the destination network, the security appliance assigns the host an IP address from the mapped pool. The translation is added only when the real host initiates the connection. The translation is in place only for the duration of the connection, and a given user does not keep the same IP address after the translation times out. For an example, see the **timeout xlate** command in the *Cisco Security Appliance Command Reference*. Users on the destination network, therefore, cannot initiate a reliable connection to a host that uses dynamic NAT, although the connection is allowed by an access list, and the security appliance rejects any attempt to connect to a real host address directly. See the "Static NAT" or "Static PAT" section for information on how to obtain reliable access to hosts.

Note In some cases, a translation is added for a connection, although the session is denied by the security appliance. This condition occurs with an outbound access list, a management-only interface, or a backup interface in which the translation times out normally. For an example, see the **show xlate** command in the *Cisco Security Appliance Command Reference*.

Figure 19-6 shows a remote host attempting to connect to the real address. The connection is denied, because the security appliance only allows returning connections to the mapped address.

Figure 19-6 Remote Host Attempts to Connect to the Real Address

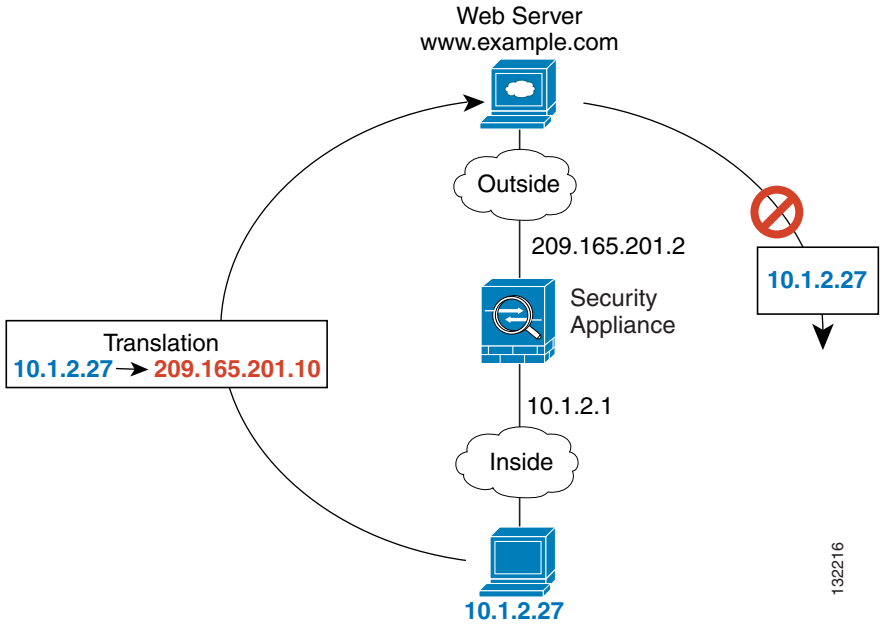
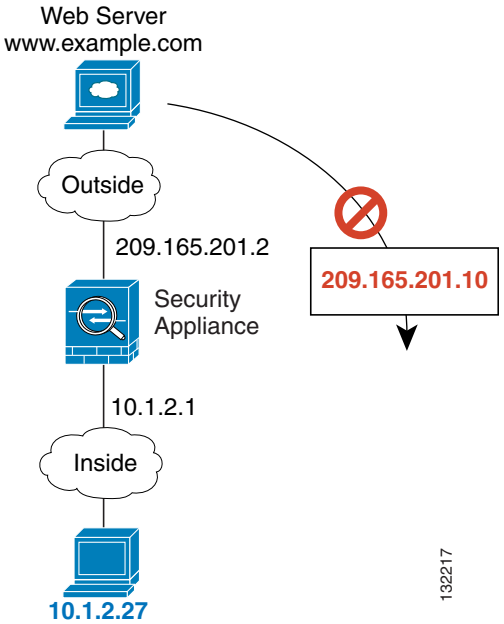


Figure 19-7 shows a remote host attempting to initiate a connection to a mapped address. This address is not currently in the translation table; therefore, the security appliance drops the packet.

Figure 19-7 Remote Host Attempts to Initiate a Connection to a Mapped Address



Note For the duration of the translation, a remote host can initiate a connection to the translated host if an access list allows it. Because the address is unpredictable, a connection to the host is unlikely. Nevertheless, in this case, you can rely on the security of the access list.

Dynamic NAT has these disadvantages:

- If the mapped pool has fewer addresses than the real group, you could run out of addresses if the amount of traffic is more than expected.

  Use PAT if this event occurs often, because PAT provides over 64,000 translations using ports of a single address.

- You have to use a large number of routable addresses in the mapped pool; if the destination network requires registered addresses, such as the Internet, you might encounter a shortage of usable addresses.

The advantage of dynamic NAT is that some protocols cannot use PAT. PAT does not work with the following:

- IP protocols that do not have a port to overload, such as GRE version 0.
- Some multimedia applications that have a data stream on one port, the control path on another port, and are not open standard.

See the "When to Use Application Protocol Inspection" section on page 26-2 for more information about NAT and PAT support.

PAT

PAT translates multiple real addresses to a single mapped IP address. Specifically, the security appliance translates the real address and source port (real socket) to the mapped address and a unique port above 1024 (mapped socket). Each connection requires a separate translation, because the source port differs for each connection. For example, 10.1.1.1:1025 requires a separate translation from 10.1.1.1:1026.

After the connection expires, the port translation also expires after 30 seconds of inactivity. The timeout is not configurable. Users on the destination network cannot reliably initiate a connection to a host that uses PAT (even if the connection is allowed by an access list). Not only can you not predict the real or mapped port number of the host, but the security appliance does not create a translation at all unless the translated host is the initiator. See the following "Static NAT" or "Static PAT" sections for reliable access to hosts.

PAT lets you use a single mapped address, thus conserving routable addresses. You can even use the security appliance interface IP address as the PAT address. PAT does not work with some multimedia applications that have a data stream that is different from the control path. See the "When to Use Application Protocol Inspection" section on page 26-2 for more information about NAT and PAT support.

Note For the duration of the translation, a remote host can initiate a connection to the translated host if an access list allows it. Because the port address (both real and mapped) is unpredictable, a connection to the host is unlikely. Nevertheless, in this case, you can rely on the security of the access list. However, policy PAT does not support time-based ACLs.

Static NAT

Static NAT creates a fixed translation of real address(es) to mapped address(es).With dynamic NAT and PAT, each host uses a different address or port for each subsequent translation. Because the mapped address is the same for each consecutive connection with static NAT, and a persistent translation rule exists, static NAT allows hosts on the destination network to initiate traffic to a translated host (if an access list exists that allows it).

The main difference between dynamic NAT and a range of addresses for static NAT is that static NAT allows a remote host to initiate a connection to a translated host (if an access list exists that allows it), while dynamic NAT does not. You also need an equal number of mapped addresses as real addresses with static NAT.

Static PAT

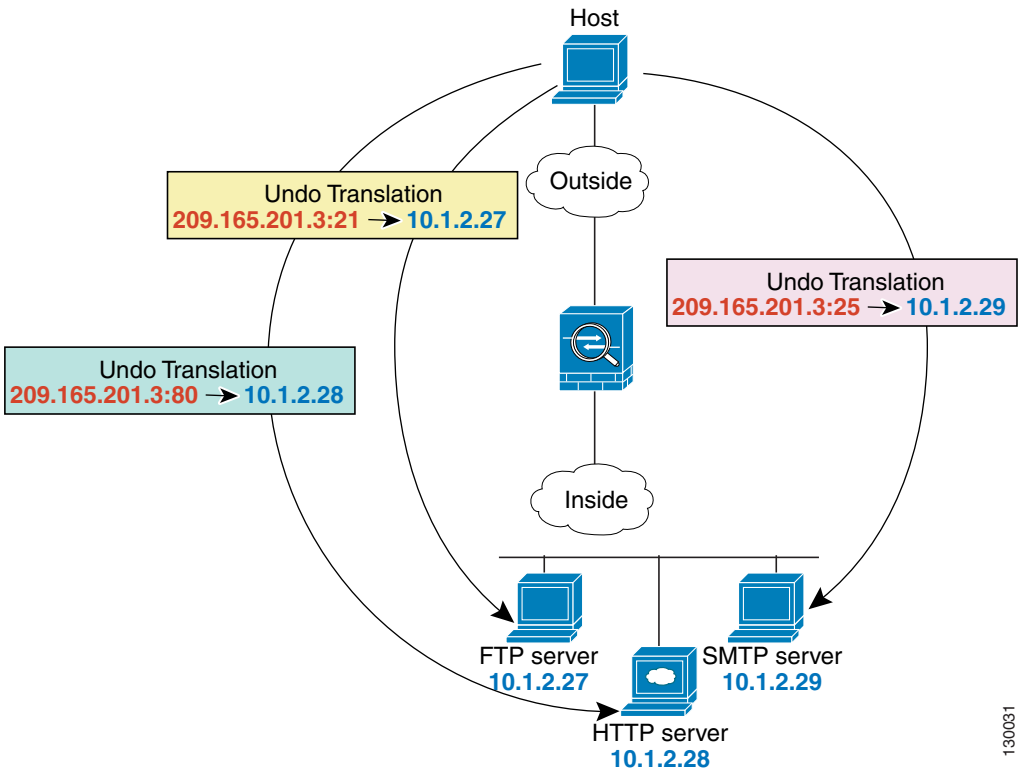
Static PAT is the same as static NAT, except that it lets you specify the protocol (TCP or UDP) and port for the real and mapped addresses.

This feature lets you identify the same mapped address across many different static statements, provided the port is different for each statement. You cannot use the same mapped address for multiple static NAT statements.

For applications that require inspection for secondary channels (for example, FTP and VoIP), the security appliance automatically translates the secondary ports.

For example, if you want to provide a single address for remote users to access FTP, HTTP, and SMTP, but these are all actually different servers on the real network, you can specify static PAT statements for each server that uses the same mapped IP address, but different ports (see Figure 19-8).

Figure 19-8 Static PAT



See the following commands for this example:

```
hostname(config)# static (inside,outside) tcp 209.165.201.3 ftp 10.1.2.27 ftp netmask
255.255.255.255
hostname(config)# static (inside,outside) tcp 209.165.201.3 http 10.1.2.28 http netmask
255.255.255.255
hostname(config)# static (inside,outside) tcp 209.165.201.3 smtp 10.1.2.29 smtp netmask
255.255.255.255
```

You can also use static PAT to translate a well-known port to a non-standard port or vice versa. For example, if inside web servers use port 8080, you can allow outside users to connect to port 80, and then undo translation to the original port 8080. Similarly, to provide extra security, you can tell web users to connect to non-standard port 6785, and then undo translation to port 80.

Bypassing NAT When NAT Control is Enabled

If you enable NAT control, then inside hosts must match a NAT rule when accessing outside hosts. If you do not want to perform NAT for some hosts, then you can bypass NAT for those hosts or you can disable NAT control. You might want to bypass NAT, for example, if you are using an application that does not support NAT. See the "When to Use Application Protocol Inspection" section on page 26-2 for information about inspection engines that do not support NAT.

You can configure traffic to bypass NAT using one of three methods. All methods achieve compatibility with inspection engines. However, each method offers slightly different capabilities, as follows:

- Identity NAT (**nat 0** command)—When you configure identity NAT (which is similar to dynamic NAT), you do not limit translation for a host on specific interfaces; you must use identity NAT for connections through all interfaces. Therefore, you cannot choose to perform normal translation on real addresses when you access interface A, but use identity NAT when accessing interface B. Regular dynamic NAT, on the other hand, lets you specify a particular interface on which to translate the addresses. Make sure that the real addresses for which you use identity NAT are routable on all networks that are available according to your access lists.

  For identity NAT, even though the mapped address is the same as the real address, you cannot initiate a connection from the outside to the inside (even if the interface access list allows it). Use static identity NAT or NAT exemption for this functionality.

- Static identity NAT (**static** command)—Static identity NAT lets you specify the interface on which you want to allow the real addresses to appear, so you can use identity NAT when you access interface A, and use regular translation when you access interface B. Static identity NAT also lets you use policy NAT, which identifies the real and destination addresses when determining the real addresses to translate (see the "Policy NAT" section on page 19-11 for more information about policy NAT). For example, you can use static identity NAT for an inside address when it accesses the outside interface and the destination is server A, but use a normal translation when accessing the outside server B.

- NAT exemption (**nat 0 access-list** command)—NAT exemption allows both translated and remote hosts to initiate connections. Like identity NAT, you do not limit translation for a host on specific interfaces; you must use NAT exemption for connections through all interfaces. However, NAT exemption does let you specify the real and destination addresses when determining the real addresses to translate (similar to policy NAT), so you have greater control using NAT exemption. However unlike policy NAT, NAT exemption does not consider the ports in the access list. NAT exemption also does not support connection settings, such as maximum TCP connections.

Policy NAT

Policy NAT lets you identify real addresses for address translation by specifying the source and destination addresses in an extended access list. You can also optionally specify the source and destination ports. Regular NAT can only consider the source addresses, and not the destination. For example, with policy NAT, you can translate the real address to mapped address A when it accesses server A, but translate the real address to mapped address B when it accesses server B.

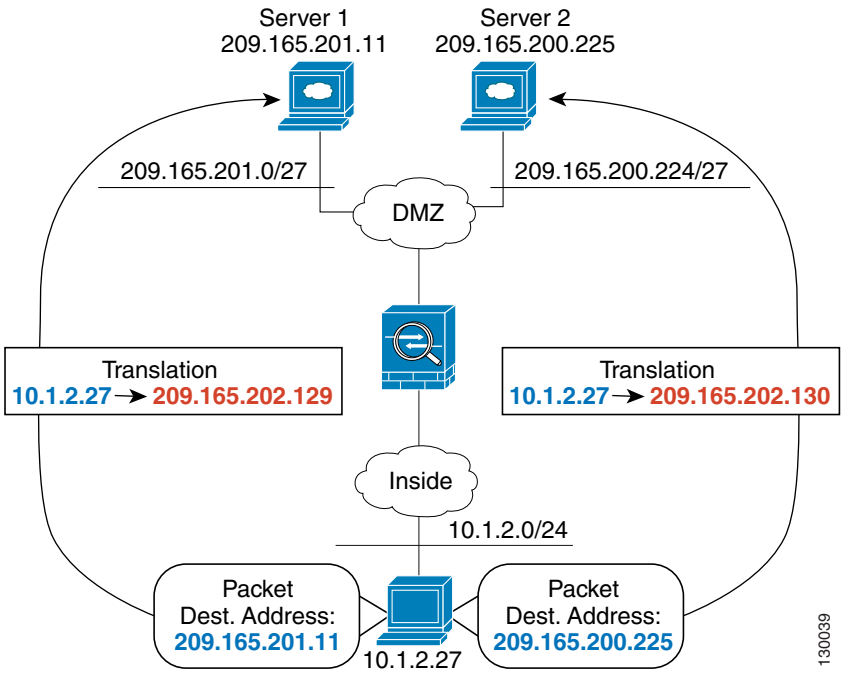
Note Policy NAT does not support time-based ACLs.

For applications that require application inspection for secondary channels (for example, FTP and VoIP), the policy specified in the policy NAT statement should include the secondary ports. When the ports cannot be predicted, the policy should specify only the IP addresses for the secondary channel. With this configuration, the security appliance translates the secondary ports.

Note All types of NAT support policy NAT, except for NAT exemption. NAT exemption uses an access list to identify the real addresses, but differs from policy NAT in that the ports are not considered. See the "Bypassing NAT" section on page 19-32 for other differences. You can accomplish the same result as NAT exemption using static identity NAT, which does support policy NAT.

Figure 19-9 shows a host on the 10.1.2.0/24 network accessing two different servers. When the host accesses the server at 209.165.201.11, the real address is translated to 209.165.202.129. When the host accesses the server at 209.165.200.225, the real address is translated to 209.165.202.130. Consequently, the host appears to be on the same network as the servers, which can help with routing.

Figure 19-9 Policy NAT with Different Destination Addresses

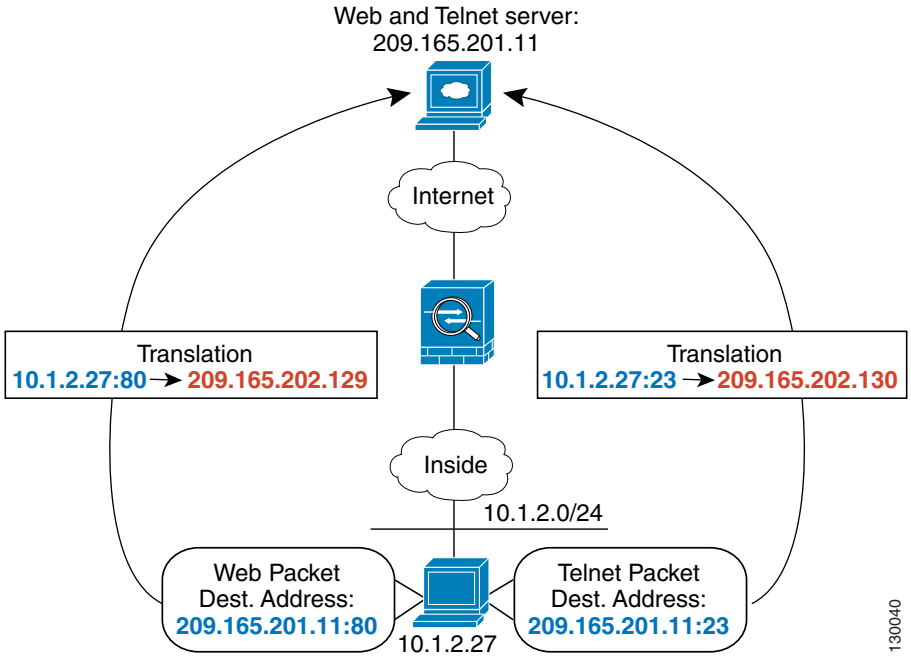


See the following commands for this example:

```
hostname(config)# access-list NET1 permit ip 10.1.2.0 255.255.255.0 209.165.201.0
255.255.255.224
hostname(config)# access-list NET2 permit ip 10.1.2.0 255.255.255.0 209.165.200.224
255.255.255.224
hostname(config)# nat (inside) 1 access-list NET1
hostname(config)# global (outside) 1 209.165.202.129
hostname(config)# nat (inside) 2 access-list NET2
hostname(config)# global (outside) 2 209.165.202.130
```

Figure 19-10 shows the use of source and destination ports. The host on the 10.1.2.0/24 network accesses a single host for both web services and Telnet services. When the host accesses the server for web services, the real address is translated to 209.165.202.129. When the host accesses the same server for Telnet services, the real address is translated to 209.165.202.130.

Figure 19-10 Policy NAT with Different Destination Ports



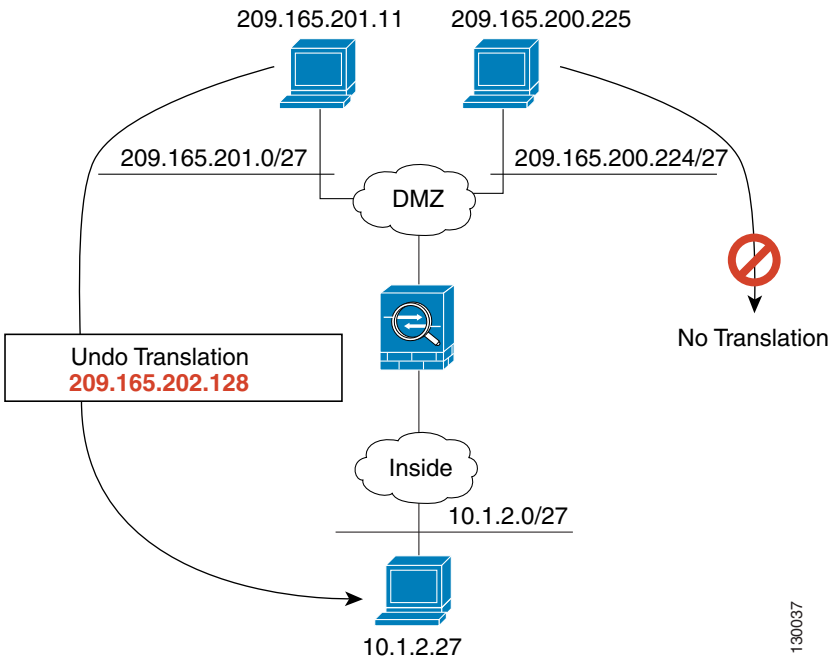
See the following commands for this example:

```
hostname(config)# access-list WEB permit tcp 10.1.2.0 255.255.255.0 209.165.201.11
255.255.255.255 eq 80
hostname(config)# access-list TELNET permit tcp 10.1.2.0 255.255.255.0 209.165.201.11
255.255.255.255 eq 23
hostname(config)# nat (inside) 1 access-list WEB
hostname(config)# global (outside) 1 209.165.202.129
hostname(config)# nat (inside) 2 access-list TELNET
hostname(config)# global (outside) 2 209.165.202.130
```

For policy static NAT (and for NAT exemption, which also uses an access list to identify traffic), both translated and remote hosts can originate traffic. For traffic originated on the translated network, the NAT access list specifies the real addresses and the *destination* addresses, but for traffic originated on the remote network, the access list identifies the real addresses and the *source* addresses of remote hosts who are allowed to connect to the host using this translation.

Figure 19-11 shows a remote host connecting to a translated host. The translated host has a policy static NAT translation that translates the real address only for traffic to and from the 209.165.201.0/27 network. A translation does not exist for the 209.165.200.224/27 network, so the translated host cannot connect to that network, nor can a host on that network connect to the translated host.

Figure 19-11 Policy Static NAT with Destination Address Translation



See the following commands for this example:

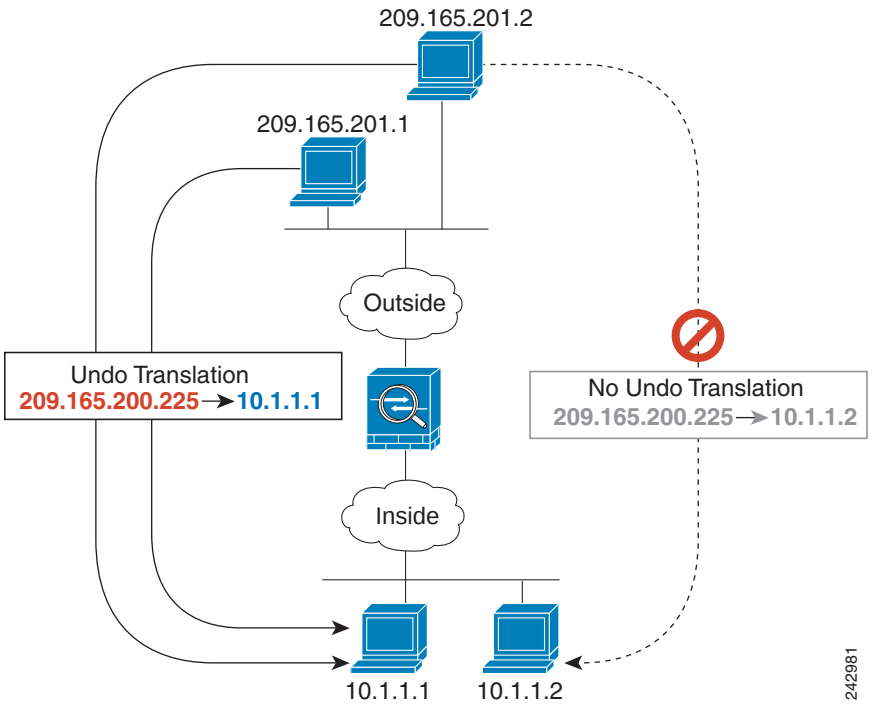
```
hostname(config)# access-list NET1 permit ip 10.1.2.0 255.255.255.224 209.165.201.0
255.255.255.224
hostname(config)# static (inside,outside) 209.165.202.128 access-list NET1
```

Note For policy static NAT, in undoing the translation, the ACL in the **static** command is not used. If the destination address in the packet matches the mapped address in the static rule, the static rule is used to untranslate the address.

Note Policy NAT does not support SQL*Net, but it is supported by regular NAT. See the "When to Use Application Protocol Inspection" section on page 26-2 for information about NAT support for other protocols.

You cannot use policy static NAT to translate different real addresses to the same mapped address. For example, Figure 19-12 shows two inside hosts, 10.1.1.1 and 10.1.1.2, that you want to be translated to 209.165.200.225. When outside host 209.165.201.1 connects to 209.165.200.225, then the connection goes to 10.1.1.1. When outside host 209.165.201.2 connects to the same mapped address, 209.165.200.225, you want the connection to go to 10.1.1.2. However, only one source address in the access list can be used. Since the first ACE is for 10.1.1.1, then all inbound connections sourced from 209.165.201.1 and 209.165.201.2 and destined to 209.165.200.255 will have their destination address translated to 10.1.1.1.

Figure 19-12 Real Addresses Cannot Share the Same Mapped Address



See the following commands for this example. (Although the second ACE in the example does allow 209.165.201.2 to connect to 209.165.200.225, it only allows 209.165.200.225 to be translated to 10.1.1.1.)

```
hostname(config)# static (in,out) 209.165.200.225 access-list policy-nat
hostname(config)# access-list policy-nat permit ip host 10.1.1.1 host 209.165.201.1
hostname(config)# access-list policy-nat permit ip host 10.1.1.2 host 209.165.201.2
```

NAT and Same Security Level Interfaces

NAT is not required between same security level interfaces even if you enable NAT control. You can optionally configure NAT if desired. However, if you configure dynamic NAT when NAT control is enabled, then NAT is required. See the "NAT Control" section on page 19-5 for more information. Also, when you specify a group of IP address(es) for dynamic NAT or PAT on a same security interface, then you must perform NAT on that group of addresses when they access any lower or same security level interface (even when NAT control is not enabled). Traffic identified for static NAT is not affected.

See the "Allowing Communication Between Interfaces on the Same Security Level" section on page 8-7 to enable same security communication.

Note The security appliance does not support VoIP inspection engines when you configure NAT on same security interfaces. These inspection engines include Skinny, SIP, and H.323. See the "When to Use Application Protocol Inspection" section on page 26-2 for supported inspection engines.

Order of NAT Commands Used to Match Real Addresses

The security appliance matches real addresses to NAT commands in the following order:

1. NAT exemption (**nat 0 access-list**)—In order, until the first match. Identity NAT is not included in this category; it is included in the regular static NAT or regular NAT category. We do not recommend overlapping addresses in NAT exemption statements because unexpected results can occur.
2. Static NAT and Static PAT (regular and policy) (**static**)—In order, until the first match. Static identity NAT is included in this category.
3. Policy dynamic NAT (**nat access-list**)—In order, until the first match. Overlapping addresses are allowed.
4. Regular dynamic NAT (**nat**)—Best match. Regular identity NAT is included in this category. The order of the NAT commands does not matter; the NAT statement that best matches the real address is used. For example, you can create a general statement to translate all addresses (0.0.0.0) on an interface. If you want to translate a subset of your network (10.1.1.1) to a different address, then you can create a statement to translate only 10.1.1.1. When 10.1.1.1 makes a connection, the specific statement for 10.1.1.1 is used because it matches the real address best. We do not recommend using overlapping statements; they use more memory and can slow the performance of the security appliance.

Mapped Address Guidelines

When you translate the real address to a mapped address, you can use the following mapped addresses:

- Addresses on the same network as the mapped interface.

  If you use addresses on the same network as the mapped interface (through which traffic exits the security appliance), the security appliance uses proxy ARP to answer any requests for mapped addresses, and thus intercepts traffic destined for a real address. This solution simplifies routing, because the security appliance does not have to be the gateway for any additional networks. However, this approach does put a limit on the number of available addresses used for translations.

  For PAT, you can even use the IP address of the mapped interface.

- Addresses on a unique network.

  If you need more addresses than are available on the mapped interface network, you can identify addresses on a different subnet. The security appliance uses proxy ARP to answer any requests for mapped addresses, and thus intercepts traffic destined for a real address. If you use OSPF to advertise mapped IP addresses that belong to a different subnet from the mapped interface, you need to create a static route to the mapped addresses that are destined to the mapped interface IP, and then redistribute this static route in OSPF. If the mapped interface is passive (not advertising routes) or you are using static routing, then you need to add a static route on the upstream router that sends traffic destined for the mapped addresses to the security appliance.

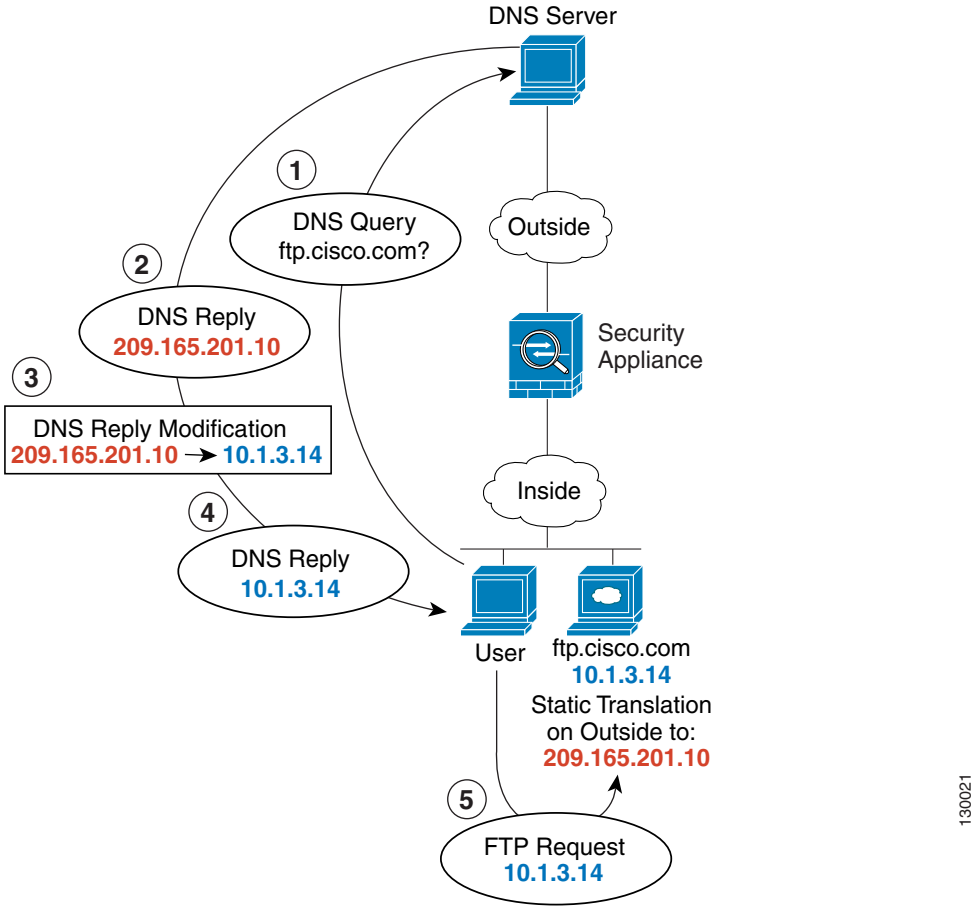
DNS and NAT

You might need to configure the security appliance to modify DNS replies by replacing the address in the reply with an address that matches the NAT configuration. You can configure DNS modification when you configure each translation.

For example, a DNS server is accessible from the outside interface. A server, ftp.cisco.com, is on the inside interface. You configure the security appliance to statically translate the ftp.cisco.com real address (10.1.3.14) to a mapped address (209.165.201.10) that is visible on the outside network (see Figure 19-13). In this case, you want to enable DNS reply modification on this static statement so that inside users who have access to ftp.cisco.com using the real address receive the real address from the DNS server, and not the mapped address.

When an inside host sends a DNS request for the address of ftp.cisco.com, the DNS server replies with the mapped address (209.165.201.10). The security appliance refers to the static statement for the inside server and translates the address inside the DNS reply to 10.1.3.14. If you do not enable DNS reply modification, then the inside host attempts to send traffic to 209.165.201.10 instead of accessing ftp.cisco.com directly.

Figure 19-13 DNS Reply Modification



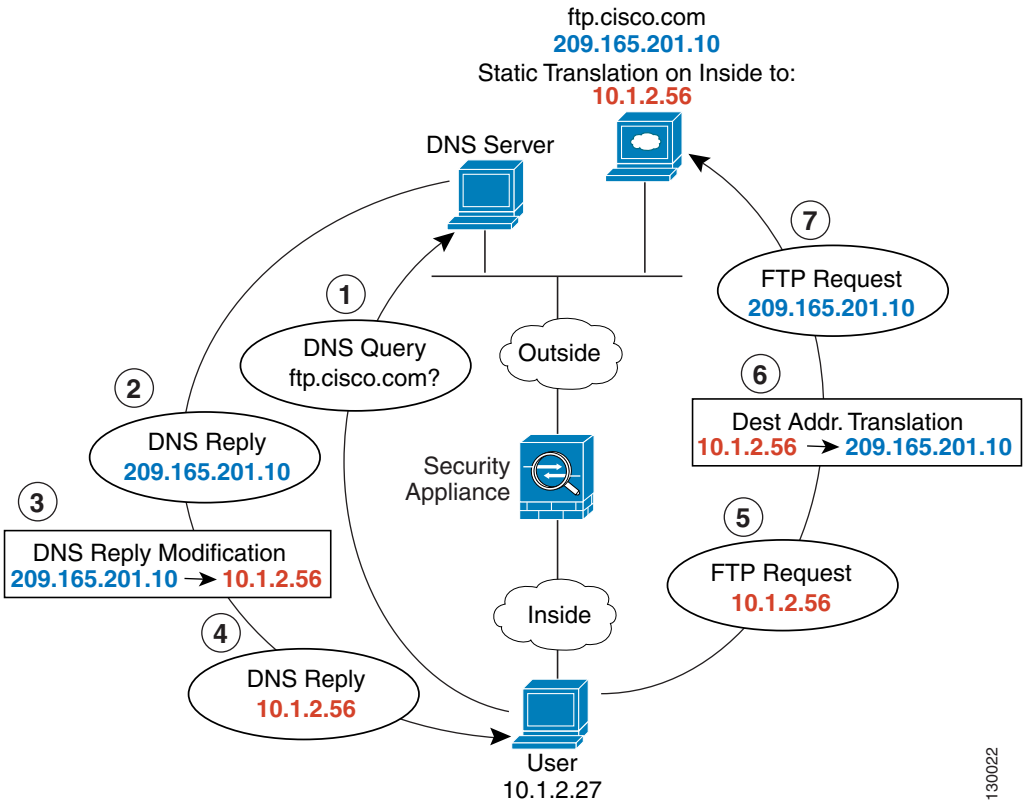
See the following command for this example:

```
hostname(config)# static (inside,outside) 209.165.201.10 10.1.3.14 netmask 255.255.255.255
dns
```

Note If a user on a different network (for example, DMZ) also requests the IP address for ftp.cisco.com from the outside DNS server, then the IP address in the DNS reply is also modified for this user, even though the user is not on the Inside interface referenced by the **static** command.

Figure 19-14 shows a web server and DNS server on the outside. The security appliance has a static translation for the outside server. In this case, when an inside user requests the address for ftp.cisco.com from the DNS server, the DNS server responds with the real address, 209.165.20.10. Because you want inside users to use the mapped address for ftp.cisco.com (10.1.2.56) you need to configure DNS reply modification for the static translation.

Figure 19-14 DNS Reply Modification Using Outside NAT



See the following command for this example:

```
hostname(config)# static (outside,inside) 10.1.2.56 209.165.201.10 netmask 255.255.255.255
dns
```

Configuring NAT Control

NAT control requires that packets traversing from an inside interface to an outside interface match a NAT rule. See the "NAT Control" section on page 19-5 for more information.

To enable NAT control, enter the following command:

```
hostname(config)# nat-control
```

To disable NAT control, enter the **no** form of the command.

Using Dynamic NAT and PAT

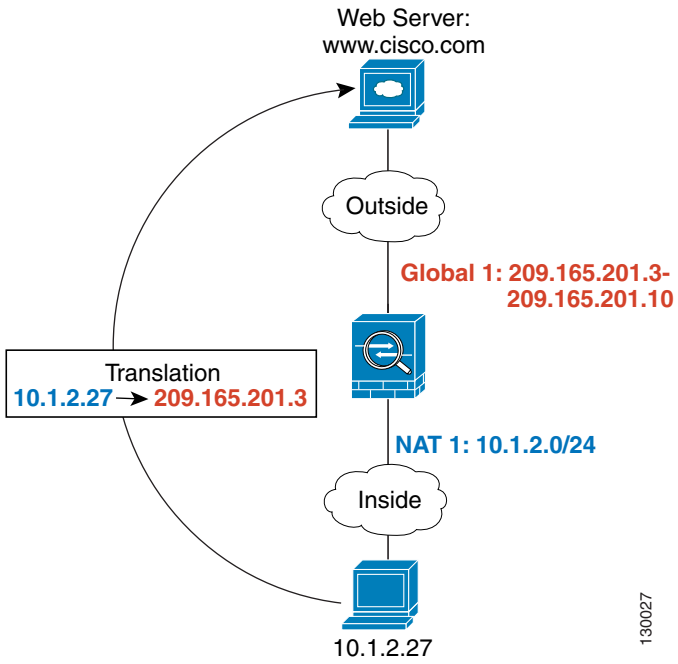
This section describes how to configure dynamic NAT and PAT, and includes the following topics:

- Dynamic NAT and PAT Implementation, page 19-19
- Configuring Dynamic NAT or PAT, page 19-25

Dynamic NAT and PAT Implementation

For dynamic NAT and PAT, you first configure a **nat** command identifying the real addresses on a given interface that you want to translate. Then you configure a separate **global** command to specify the mapped addresses when exiting another interface (in the case of PAT, this is one address). Each **nat** command matches a **global** command by comparing the NAT ID, a number that you assign to each command (see Figure 19-15).

Figure 19-15 nat and global ID Matching

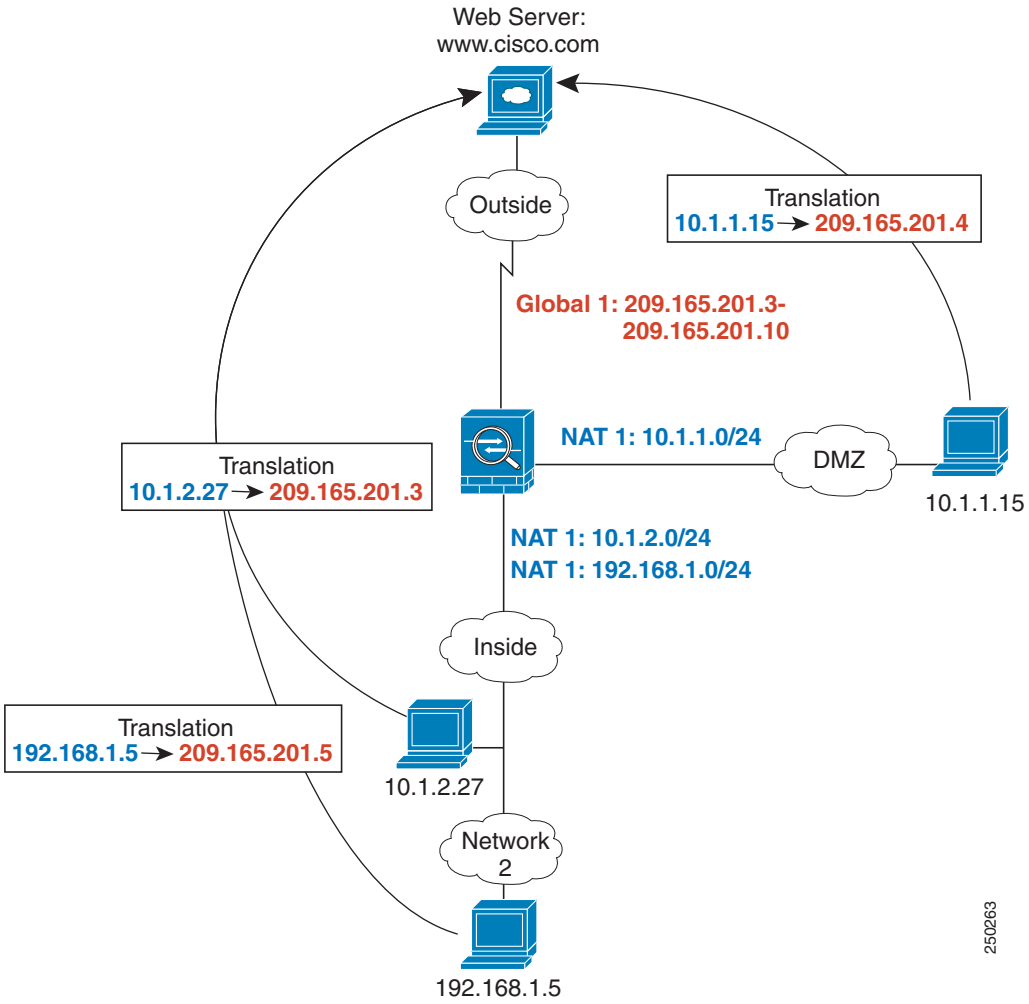


See the following commands for this example:

```
hostname(config)# nat (inside) 1 10.1.2.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.3-209.165.201.10
```

You can enter multiple **nat** commands using the same NAT ID on one or more interfaces; they all use the same **global** command when traffic exits a given interface. For example, you can configure **nat** commands for Inside and DMZ interfaces, both on NAT ID 1. Then you configure a **global** command on the Outside interface that is also on ID 1. Traffic from the Inside interface and the DMZ interface share a mapped pool or a PAT address when exiting the Outside interface (see Figure 19-16).

Figure 19-16 nat Commands on Multiple Interfaces

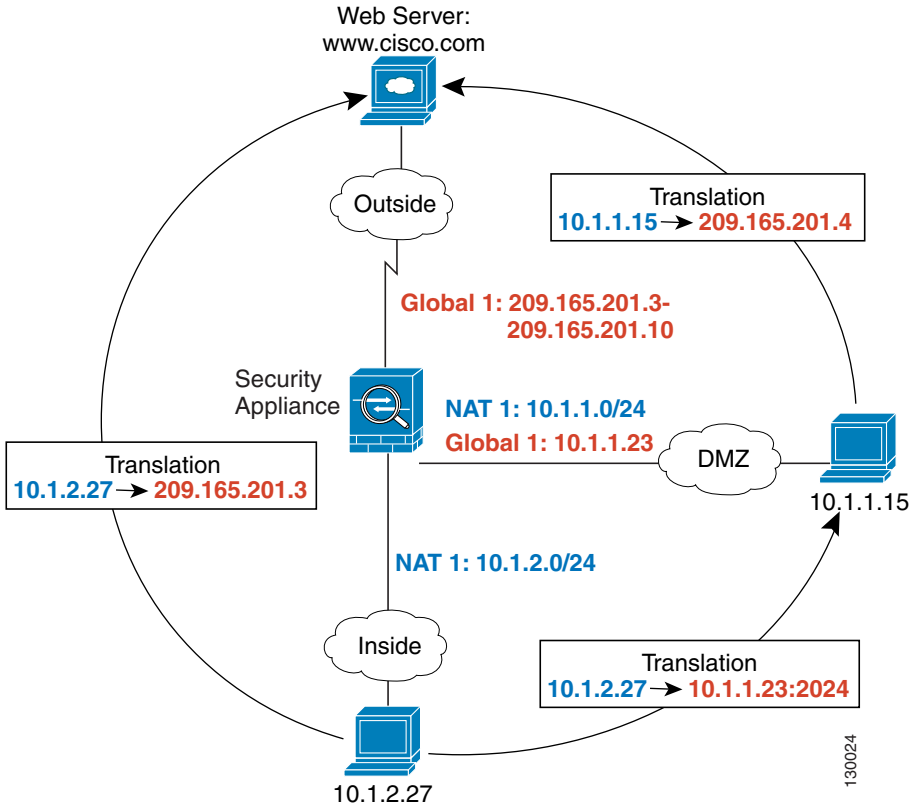


See the following commands for this example:

```
hostname(config)# nat (inside) 1 10.1.2.0 255.255.255.0
hostname(config)# nat (inside) 1 192.168.1.0 255.255.255.0
hostname(config)# nat (dmz) 1 10.1.1.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.3-209.165.201.10
```

You can also enter a **global** command for each interface using the same NAT ID. If you enter a **global** command for the Outside and DMZ interfaces on ID 1, then the Inside **nat** command identifies traffic to be translated when going to both the Outside and the DMZ interfaces. Similarly, if you also enter a **nat** command for the DMZ interface on ID 1, then the **global** command on the Outside interface is also used for DMZ traffic. (See Figure 19-17).

Figure 19-17 global and nat Commands on Multiple Interfaces

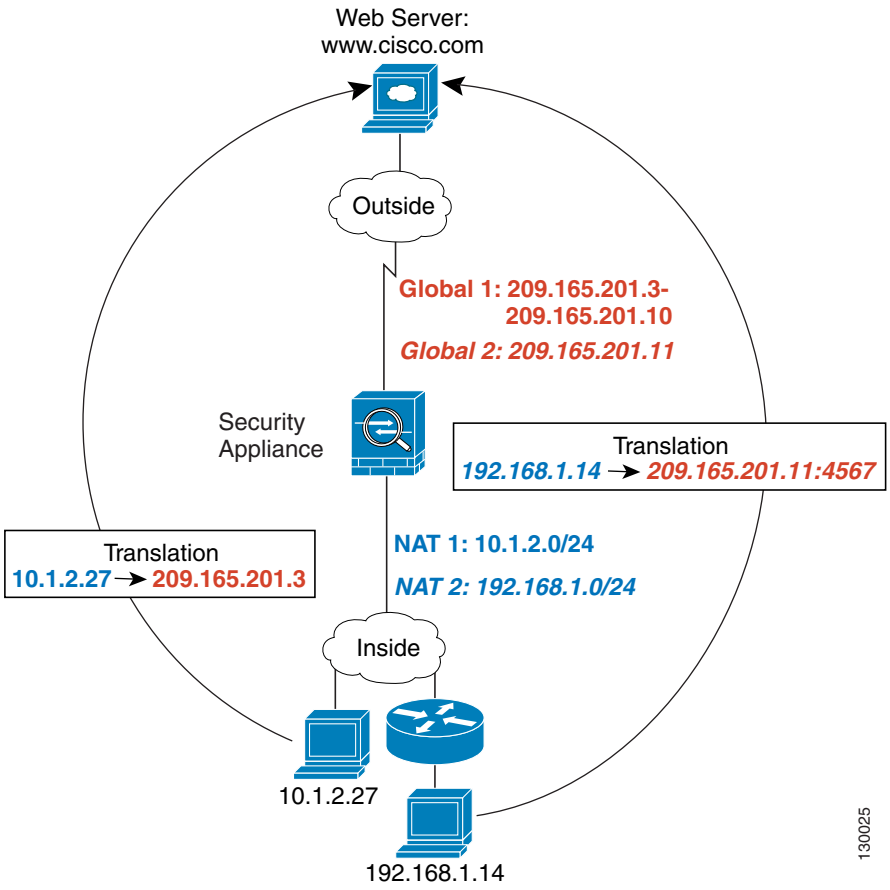


See the following commands for this example:

```
hostname(config)# nat (inside) 1 10.1.2.0 255.255.255.0
hostname(config)# nat (dmz) 1 10.1.1.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.3-209.165.201.10
hostname(config)# global (dmz) 1 10.1.1.23
```

If you use different NAT IDs, you can identify different sets of real addresses to have different mapped addresses. For example, on the Inside interface, you can have two **nat** commands on two different NAT IDs. On the Outside interface, you configure two **global** commands for these two IDs. Then, when traffic from Inside network A exits the Outside interface, the IP addresses are translated to pool A addresses; while traffic from Inside network B are translated to pool B addresses (see Figure 19-18). If you use policy NAT, you can specify the same real addresses for multiple **nat** commands, as long as the the destination addresses and ports are unique in each access list.

Figure 19-18 Different NAT IDs

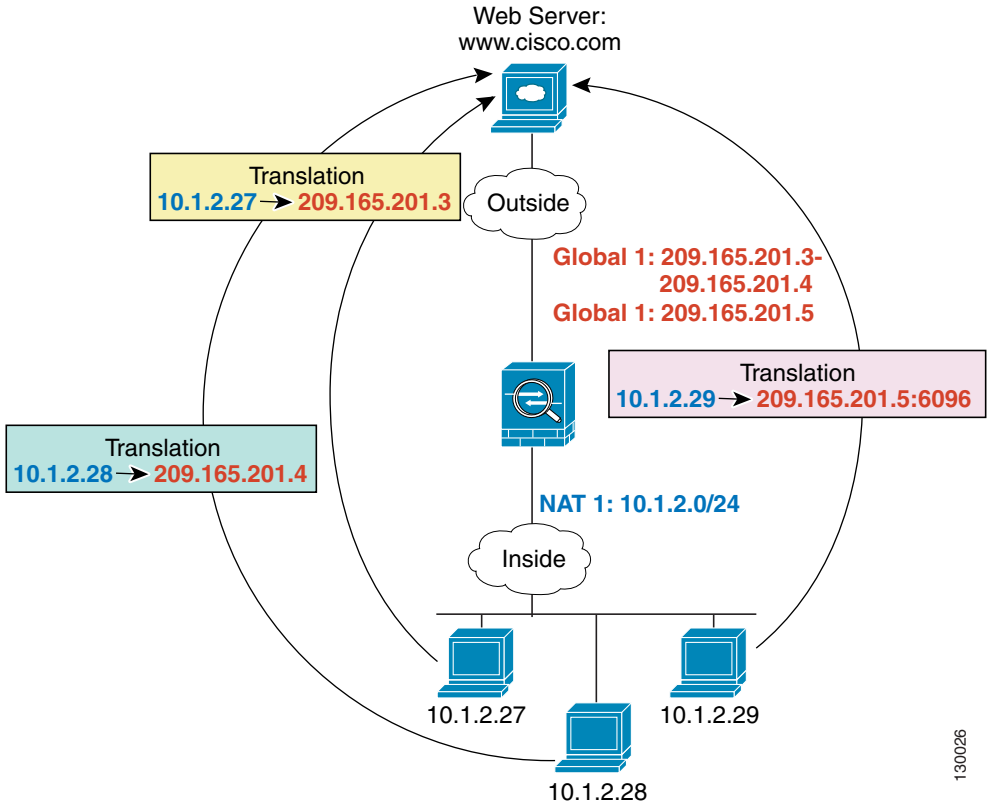


See the following commands for this example:

```
hostname(config)# nat (inside) 1 10.1.2.0 255.255.255.0
hostname(config)# nat (inside) 2 192.168.1.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.3-209.165.201.10
hostname(config)# global (outside) 2 209.165.201.11
```

You can enter multiple **global** commands for one interface using the same NAT ID; the security appliance uses the dynamic NAT **global** commands first, in the order they are in the configuration, and then uses the PAT **global** commands in order. You might want to enter both a dynamic NAT **global** command and a PAT **global** command if you need to use dynamic NAT for a particular application, but want to have a backup PAT statement in case all the dynamic NAT addresses are depleted. Similarly, you might enter two PAT statements if you need more than the approximately 64,000 PAT sessions that a single PAT mapped statement supports (see Figure 19-19).

Figure 19-19 NAT and PAT Together

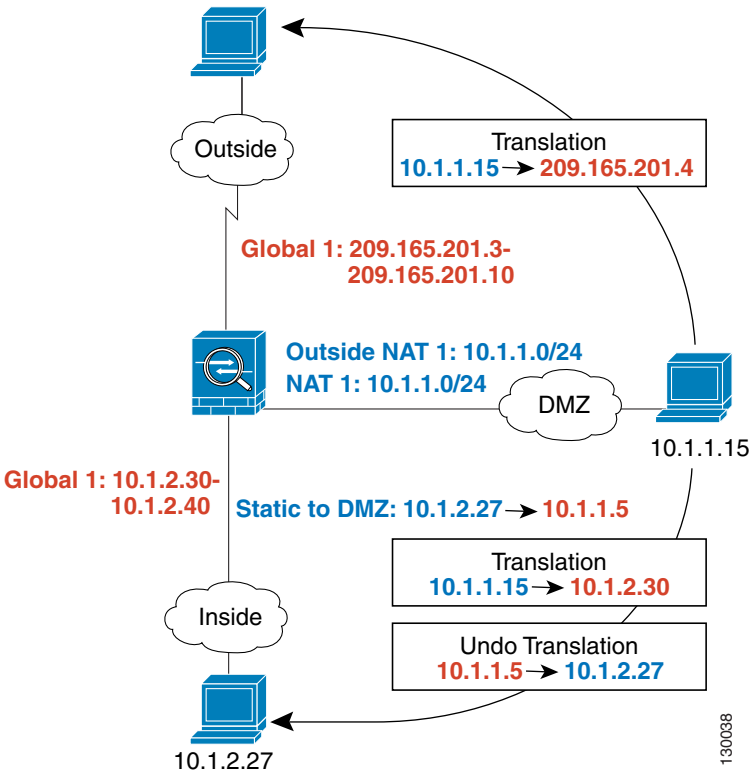


See the following commands for this example:

```
hostname(config)# nat (inside) 1 10.1.2.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.3-209.165.201.4
hostname(config)# global (outside) 1 209.165.201.5
```

For outside NAT (from outside to inside), you need to use the **outside** keyword in the **nat** command. If you also want to translate the same traffic when it accesses an outside interface (for example, traffic on a DMZ is translated when accessing the Inside and the Outside interfaces), then you must configure a separate **nat** command without the **outside** option. In this case, you can identify the same addresses in both statements and use the same NAT ID (see Figure 19-20). Note that for outside NAT (DMZ interface to Inside interface), the inside host uses a **static** command to allow outside access, so both the source and destination addresses are translated.

Figure 19-20 Outside NAT and Inside NAT Combined



See the following commands for this example:

```
hostname(config)# nat (dmz) 1 10.1.1.0 255.255.255.0 outside
hostname(config)# nat (dmz) 1 10.1.1.0 255.255.255.0
hostname(config)# static (inside,dmz) 10.1.1.5 10.1.2.27 netmask 255.255.255.255
hostname(config)# global (outside) 1 209.165.201.3-209.165.201.4
hostname(config)# global (inside) 1 10.1.2.30-1-10.1.2.40
```

When you specify a group of IP address(es) in a **nat** command, then you must perform NAT on that group of addresses when they access any lower or same security level interface; you must apply a **global** command with the same NAT ID on each interface, or use a **static** command. NAT is not required for that group when it accesses a higher security interface, because to perform NAT from outside to inside, you must create a separate **nat** command using the **outside** keyword. If you do apply outside NAT, then the NAT requirements preceding come into effect for that group of addresses when they access all higher security interfaces. Traffic identified by a **static** command is not affected.

Configuring Dynamic NAT or PAT

This section describes how to configure dynamic NAT or dynamic PAT. The configuration for dynamic NAT and PAT are almost identical; for NAT you specify a range of mapped addresses, and for PAT you specify a single address.

Figure 19-21 shows a typical dynamic NAT scenario. Only translated hosts can create a NAT session, and responding traffic is allowed back. The mapped address is dynamically assigned from a pool defined by the **global** command.

Figure 19-21 Dynamic NAT

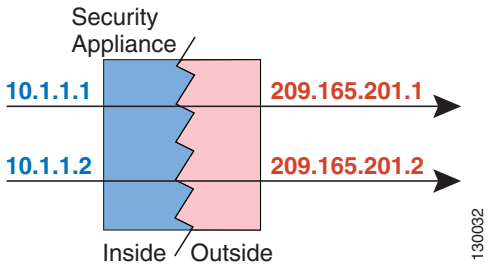
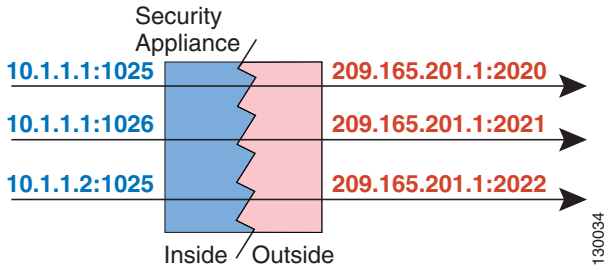


Figure 19-22 shows a typical dynamic PAT scenario. Only translated hosts can create a NAT session, and responding traffic is allowed back. The mapped address defined by the **global** command is the same for each translation, but the port is dynamically assigned.

Figure 19-22 Dynamic PAT



For more information about dynamic NAT, see the "Dynamic NAT" section on page 19-6. For more information about PAT, see the "PAT" section on page 19-8.

Note If you change the NAT configuration, and you do not want to wait for existing translations to time out before the new NAT information is used, you can clear the translation table using the **clear xlate** command. However, clearing the translation table disconnects all current connections that use translations.

To configure dynamic NAT or PAT, perform the following steps:

Step 1 To identify the real addresses that you want to translate, enter one of the following commands:

- Policy NAT:

```
hostname(config)# nat (real_interface) nat_id access-list acl_name [dns] [outside]
[norandomseq] [[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns]
```

You can identify overlapping addresses in other **nat** commands. For example, you can identify 10.1.1.0 in one command, but 10.1.1.1 in another. The traffic is matched to a policy NAT command in order, until the first match, or for regular NAT, using the best match.

The options for this command are as follows:

  - **access-list** *acl_name*—Identify the real addresses and destination addresses using an extended access list. Create the extended access list using the **access-list extended** command (see the "Adding an Extended Access List" section on page 18-5). This access list should include only **permit** ACEs. You can optionally specify the real and destination ports in the access list using the **eq** operator. Policy NAT considers the **inactive** and **time-range** keywords, but it does not support ACL with all **inactive** and **time-range** ACEs.
  - *nat_id*—An integer between 1 and 65535. The NAT ID should match a **global** command NAT ID. See the "Dynamic NAT and PAT Implementation" section on page 19-19 for more information about how NAT IDs are used. **0** is reserved for NAT exemption. (See the "Configuring NAT Exemption" section on page 19-35 for more information about NAT exemption.)
  - **dns**—If your **nat** command includes the address of a host that has an entry in a DNS server, and the DNS server is on a different interface from a client, then the client and the DNS server need different addresses for the host; one needs the mapped address and one needs the real address. This option rewrites the address in the DNS reply to the client. The translated host needs to be on the same interface as either the client or the DNS server. Typically, hosts that need to allow access from other interfaces use a static translation, so this option is more likely to be used with the **static** command. (See the "DNS and NAT" section on page 19-17 for more information.)
  - **outside**—If this interface is on a lower security level than the interface you identify by the matching **global** statement, then you must enter **outside** to identify the NAT instance as outside NAT.
  - **norandomseq**, **tcp** *tcp_max_conns*, **udp** *udp_max_conns*, and *emb_limit*—These keywords set connection limits. However, we recommend using a more versatile method for setting connection limits; see the "Configuring Connection Limits and Timeouts" section on page 24-17.

- Regular NAT:

```
hostname(config)# nat (real_interface) nat_id real_ip [mask [dns] [outside]
[norandomseq] [[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns]]
```

The *nat_id* argument is an integer between 1 and 2147483647. The NAT ID must match a **global** command NAT ID. See the "Dynamic NAT and PAT Implementation" section on page 19-19 for more information about how NAT IDs are used. **0** is reserved for identity NAT. See the "Configuring Identity NAT" section on page 19-32 for more information about identity NAT.

See the preceding policy NAT command for information about other options.

Step 2 To identify the mapped address(es) to which you want to translate the real addresses when they exit a particular interface, enter the following command:

```
hostname(config)# global (mapped_interface) nat_id {mapped_ip[-mapped_ip] | interface}
```

This NAT ID should match a **nat** command NAT ID. The matching **nat** command identifies the addresses that you want to translate when they exit this interface.

You can specify a single address (for PAT) or a range of addresses (for NAT). The range can go across subnet boundaries if desired. For example, you can specify the following "supernet":

```
192.168.1.1-192.168.2.254
```

For example, to translate the 10.1.1.0/24 network on the inside interface, enter the following command:

```
hostname(config)# nat (inside) 1 10.1.1.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.1-209.165.201.30
```

To identify a pool of addresses for dynamic NAT as well as a PAT address for when the NAT pool is exhausted, enter the following commands:

```
hostname(config)# nat (inside) 1 10.1.1.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.5
hostname(config)# global (outside) 1 209.165.201.10-209.165.201.20
```

To translate the lower security dmz network addresses so they appear to be on the same network as the inside network (10.1.1.0), for example, to simplify routing, enter the following commands:

```
hostname(config)# nat (dmz) 1 10.1.2.0 255.255.255.0 outside dns
hostname(config)# global (inside) 1 10.1.1.45
```

To identify a single real address with two different destination addresses using policy NAT, enter the following commands (see Figure 19-9 on page 19-12 for a related figure):

```
hostname(config)# access-list NET1 permit ip 10.1.2.0 255.255.255.0 209.165.201.0
255.255.255.224
hostname(config)# access-list NET2 permit ip 10.1.2.0 255.255.255.0 209.165.200.224
255.255.255.224
hostname(config)# nat (inside) 1 access-list NET1 tcp 0 2000 udp 10000
hostname(config)# global (outside) 1 209.165.202.129
hostname(config)# nat (inside) 2 access-list NET2 tcp 1000 500 udp 2000
hostname(config)# global (outside) 2 209.165.202.130
```

To identify a single real address/destination address pair that use different ports using policy NAT, enter the following commands (see Figure 19-10 on page 19-13 for a related figure):

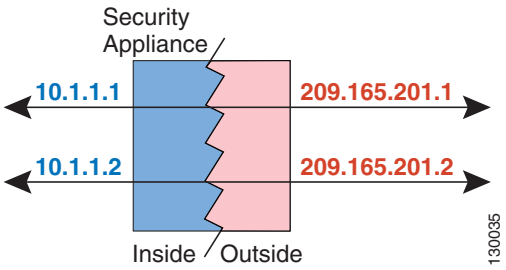
```
hostname(config)# access-list WEB permit tcp 10.1.2.0 255.255.255.0 209.165.201.11
255.255.255.255 eq 80
hostname(config)# access-list TELNET permit tcp 10.1.2.0 255.255.255.0 209.165.201.11
255.255.255.255 eq 23
hostname(config)# nat (inside) 1 access-list WEB
hostname(config)# global (outside) 1 209.165.202.129
hostname(config)# nat (inside) 2 access-list TELNET
hostname(config)# global (outside) 2 209.165.202.130
```

Using Static NAT

This section describes how to configure a static translation.

Figure 19-23 shows a typical static NAT scenario. The translation is always active so both translated and remote hosts can originate connections, and the mapped address is statically assigned by the **static** command.

Figure 19-23 Static NAT



You cannot use the same real or mapped address in multiple **static** commands between the same two interfaces unless you use static PAT (see the "Using Static PAT" section on page 19-29). Do not use a mapped address in the **static** command that is also defined in a **global** command for the same mapped interface.

For more information about static NAT, see the "Static NAT" section on page 19-9.

Note If you remove a **static** command, existing connections that use the translation are not affected. To remove these connections, enter the **clear local-host** command.

You cannot clear static translations from the translation table with the **clear xlate** command; you must remove the **static** command instead. Only dynamic translations created by the **nat** and **global** commands can be removed with the **clear xlate** command.

To configure static NAT, enter one of the following commands.

- For policy static NAT, enter the following command:

```
hostname(config)# static (real_interface,mapped_interface) {mapped_ip | interface}
access-list acl_name [dns] [norandomseq] [[tcp] tcp_max_conns [emb_limit]]
[udp udp_max_conns]
```

  Identify the real addresses and destination/source addresses using an extended access list. Create the extended access list using the **access-list extended** command (see the "Adding an Extended Access List" section on page 18-5). The first address in the access list is the real address; the second address is either the source or destiniation address, depending on where the traffic originates. For example, to translate the real address 10.1.1.1 to the mapped address 192.168.1.1 when 10.1.1.1 sends traffic to the 209.165.200.224 network, the **access-list** and **static** commands are:

```
hostname(config)# access-list TEST extended ip host 10.1.1.1 209.165.200.224
255.255.255.224
hostname(config)# static (inside,outside) 192.168.1.1 access-list TEST
```

In this case, the second address is the destination address. However, the same configuration is used for hosts to originate a connection to the mapped address. For example, when a host on the 209.165.200.224/27 network initiates a connection to 192.168.1.1, then the second address in the access list is the source address.

This access list should include only **permit** ACEs. You can optionally specify the real and destination ports in the access list using the **eq** operator. Policy NAT does not consider the **inactive** or **time-range** keywords; all ACEs are considered to be active for policy NAT configuration. See the "Policy NAT" section on page 19-11 for more information.

If you specify a network for translation (for example, 10.1.1.0 255.255.255.0), then the security appliance translates the .0 and .255 addresses. If you want to prevent access to these addresses, be sure to configure an access list to deny access.

See the "Configuring Dynamic NAT or PAT" section on page 19-25 for information about the other options.

- To configure regular static NAT, enter the following command:

```
hostname(config)# static (real_interface,mapped_interface) {mapped_ip | interface}
real_ip [netmask mask] [dns] [norandomseq] [[tcp] tcp_max_conns [emb_limit]]
[udp udp_max_conns]
```

See the "Configuring Dynamic NAT or PAT" section on page 19-25 for information about the options.

For example, the following policy static NAT example shows a single real address that is translated to two mapped addresses depending on the destination address (see Figure 19-9 on page 19-12 for a related figure):

```
hostname(config)# access-list NET1 permit ip host 10.1.2.27 209.165.201.0 255.255.255.224
hostname(config)# access-list NET2 permit ip host 10.1.2.27 209.165.200.224
255.255.255.224
hostname(config)# static (inside,outside) 209.165.202.129 access-list NET1
hostname(config)# static (inside,outside) 209.165.202.130 access-list NET2
```

The following command maps an inside IP address (10.1.1.3) to an outside IP address (209.165.201.12):

```
hostname(config)# static (inside,outside) 209.165.201.12 10.1.1.3 netmask 255.255.255.255
```

The following command maps the outside address (209.165.201.15) to an inside address (10.1.1.6):

```
hostname(config)# static (outside,inside) 10.1.1.6 209.165.201.15 netmask 255.255.255.255
```

The following command statically maps an entire subnet:

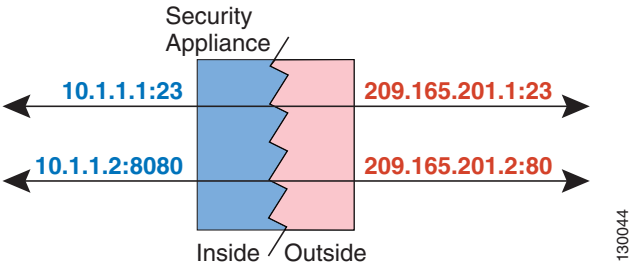
```
hostname(config)# static (inside,dmz) 10.1.1.0 10.1.2.0 netmask 255.255.255.0
```

Using Static PAT

This section describes how to configure a static port translation. Static PAT lets you translate the real IP address to a mapped IP address, as well as the real port to a mapped port. You can choose to translate the real port to the same port, which lets you translate only specific types of traffic, or you can take it further by translating to a different port.

Figure 19-24 shows a typical static PAT scenario. The translation is always active so both translated and remote hosts can originate connections, and the mapped address and port is statically assigned by the **static** command.

Figure 19-24 Static PAT



For applications that require application inspection for secondary channels (for example, FTP and VoIP), the security appliance automatically translates the secondary ports.

Do not use a mapped address in the **static** command that is also defined in a **global** command for the same mapped interface.

For more information about static PAT, see the "Static PAT" section on page 19-9.

Note If you remove a **static** command, existing connections that use the translation are not affected. To remove these connections, enter the **clear local-host** command.

You cannot clear static translations from the translation table with the **clear xlate** command; you must remove the **static** command instead. Only dynamic translations created by the **nat** and **global** commands can be removed with the **clear xlate** command.

To configure static PAT, enter one of the following commands.

- For policy static PAT, enter the following command:

```
hostname(config)# static (real_interface,mapped_interface) {tcp | udp}
{mapped_ip | interface} mapped_port access-list acl_name [dns] [norandomseq]
[[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns]
```

  Identify the real addresses and destination/source addresses using an extended access list. Create the extended access list using the **access-list extended** command (see the "Adding an Extended Access List" section on page 18-5). The protocol in the access list must match the protocol you set in this command. For example, if you specify **tcp** in the **static** command, then you must specify **tcp** in the access list. Specify the port using the **eq** operator.

  The first address in the access list is the real address; the second address is either the source or destiniation address, depending on where the traffic originates. For example, to translate the real address 10.1.1.1/Telnet to the mapped address 192.168.1.1/Telnet when 10.1.1.1 sends traffic to the 209.165.200.224 network, the **access-list** and **static** commands are:

```
hostname(config)# access-list TEST extended tcp host 10.1.1.1 eq telnet
209.165.200.224 255.255.255.224
hostname(config)# static (inside,outside) tcp 192.168.1.1 telnet access-list TEST
```

In this case, the second address is the destination address. However, the same configuration is used for hosts to originate a connection to the mapped address. For example, when a host on the 209.165.200.224 network initiates a Telnet connection to 192.168.1.1, then the second address in the access list is the source address.

This access list should include only **permit** ACEs. Policy NAT does not consider the **inactive** or **time-range** keywords; all ACEs are considered to be active for policy NAT configuration. See the "Policy NAT" section on page 19-11 for more information.

If you specify a network for translation (for example, 10.1.1.0 255.255.255.0), then the security appliance translates the .0 and .255 addresses. If you want to prevent access to these addresses, be sure to configure an access list to deny access.

See the "Configuring Dynamic NAT or PAT" section on page 19-25 for information about the other options.

- To configure regular static PAT, enter the following command:

```
hostname(config)# static (real_interface,mapped_interface) {tcp | udp} {mapped_ip |
interface} mapped_port real_ip real_port [netmask mask] [dns] [norandomseq] [[tcp]
tcp_max_conns [emb_limit]] [udp udp_max_conns]
```

See the "Configuring Dynamic NAT or PAT" section on page 19-25 for information about the options.

Note When configuring static PAT with FTP, you need to add entries for both TCP ports 20 and 21. You must specify port 20 so that the source port for the active transfer is not modified to another port, which may interfere with other devices that perform NAT on FTP traffic.

For example, for Telnet traffic initiated from hosts on the 10.1.3.0 network to the security appliance outside interface (10.1.2.14), you can redirect the traffic to the inside host at 10.1.1.15 by entering the following commands:

```
hostname(config)# access-list TELNET permit tcp host 10.1.1.15 eq telnet 10.1.3.0
255.255.255.0
hostname(config)# static (inside,outside) tcp 10.1.2.14 telnet access-list TELNET
```

For HTTP traffic initiated from hosts on the 10.1.3.0 network to the security appliance outside interface (10.1.2.14), you can redirect the traffic to the inside host at 10.1.1.15 by entering:

```
hostname(config)# access-list HTTP permit tcp host 10.1.1.15 eq http 10.1.3.0
255.255.255.0
hostname(config)# static (inside,outside) tcp 10.1.2.14 http access-list HTTP
```

To redirect Telnet traffic from the security appliance outside interface (10.1.2.14) to the inside host at 10.1.1.15, enter the following command:

```
hostname(config)# static (inside,outside) tcp 10.1.2.14 telnet 10.1.1.15 telnet netmask
255.255.255.255
```

If you want to allow the preceding real Telnet server to initiate connections, though, then you need to provide additional translation. For example, to translate all other types of traffic, enter the following commands. The original **static** command provides translation for Telnet to the server, while the **nat** and **global** commands provide PAT for outbound connections from the server.

```
hostname(config)# static (inside,outside) tcp 10.1.2.14 telnet 10.1.1.15 telnet netmask
255.255.255.255
hostname(config)# nat (inside) 1 10.1.1.15 255.255.255.255
hostname(config)# global (outside) 1 10.1.2.14
```

If you also have a separate translation for all inside traffic, and the inside hosts use a different mapped address from the Telnet server, you can still configure traffic initiated from the Telnet server to use the same mapped address as the **static** statement that allows Telnet traffic to the server. You need to create a more exclusive **nat** statement just for the Telnet server. Because **nat** statements are read for the best match, more exclusive **nat** statements are matched before general statements. The following example shows the Telnet **static** statement, the more exclusive **nat** statement for initiated traffic from the Telnet server, and the statement for other inside hosts, which uses a different mapped address.

```
hostname(config)# static (inside,outside) tcp 10.1.2.14 telnet 10.1.1.15 telnet netmask
255.255.255.255
hostname(config)# nat (inside) 1 10.1.1.15 255.255.255.255
hostname(config)# global (outside) 1 10.1.2.14
hostname(config)# nat (inside) 2 10.1.1.0 255.255.255.0
hostname(config)# global (outside) 2 10.1.2.78
```

To translate a well-known port (80) to another port (8080), enter the following command:

```
hostname(config)# static (inside,outside) tcp 10.1.2.45 80 10.1.1.16 8080 netmask
255.255.255.255
```

Bypassing NAT

This section describes how to bypass NAT. You might want to bypass NAT when you enable NAT control. You can bypass NAT using identity NAT, static identity NAT, or NAT exemption. See the "Bypassing NAT When NAT Control is Enabled" section on page 19-10 for more information about these methods. This section includes the following topics:

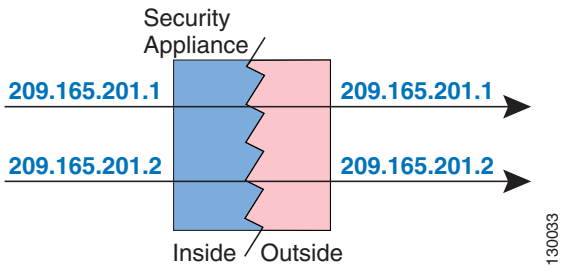
- Configuring Identity NAT, page 19-32
- Configuring Static Identity NAT, page 19-33
- Configuring NAT Exemption, page 19-35

Configuring Identity NAT

Identity NAT translates the real IP address to the same IP address. Only "translated" hosts can create NAT translations, and responding traffic is allowed back.

Figure 19-25 shows a typical identity NAT scenario.

Figure 19-25 Identity NAT



Note If you change the NAT configuration, and you do not want to wait for existing translations to time out before the new NAT information is used, you can clear the translation table using the **clear xlate** command. However, clearing the translation table disconnects all current connections that use translations.

To configure identity NAT, enter the following command:

```
hostname(config)# nat (real_interface) 0 real_ip [mask [dns] [outside] [norandomseq]
[[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns]
```

See the "Configuring Dynamic NAT or PAT" section on page 19-25 for information about the options.

For example, to use identity NAT for the inside 10.1.1.0/24 network, enter the following command:

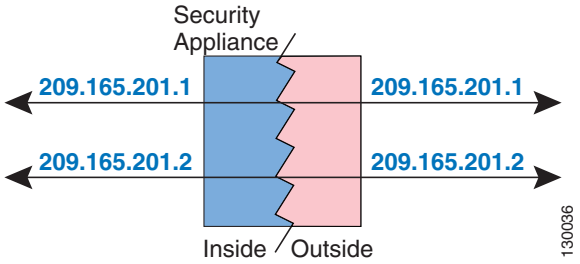
```
hostname(config)# nat (inside) 0 10.1.1.0 255.255.255.0
```

Configuring Static Identity NAT

Static identity NAT translates the real IP address to the same IP address. The translation is always active, and both "translated" and remote hosts can originate connections. Static identity NAT lets you use regular NAT or policy NAT. Policy NAT lets you identify the real and destination addresses when determining the real addresses to translate (see the "Policy NAT" section on page 19-11 for more information about policy NAT). For example, you can use policy static identity NAT for an inside address when it accesses the outside interface and the destination is server A, but use a normal translation when accessing the outside server B.

Figure 19-26 shows a typical static identity NAT scenario.

Figure 19-26 Static Identity NAT



Note If you remove a **static** command, existing connections that use the translation are not affected. To remove these connections, enter the **clear local-host** command.

You cannot clear static translations from the translation table with the **clear xlate** command; you must remove the **static** command instead. Only dynamic translations created by the **nat** and **global** commands can be removed with the **clear xlate** command.

To configure static identity NAT, enter one of the following commands:

- To configure policy static identity NAT, enter the following command:

  ```
  hostname(config)# static (real_interface,mapped_interface) real_ip access-list acl_id
  [dns] [norandomseq] [[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns]
  ```

  Create the extended access list using the **access-list extended** command (see the "Adding an Extended Access List" section on page 18-5). This access list should include only **permit** ACEs. Make sure the source address in the access list matches the *real_ip* in this command. Policy NAT does not consider the **inactive** or **time-range** keywords; all ACEs are considered to be active for policy NAT configuration. See the "Policy NAT" section on page 19-11 for more information.

  See the "Configuring Dynamic NAT or PAT" section on page 19-25 for information about the other options.

- To configure regular static identity NAT, enter the following command:

  ```
  hostname(config)# static (real_interface,mapped_interface) real_ip real_ip [netmask
  mask] [dns] [norandomseq] [[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns]
  ```

  Specify the same IP address for both *real_ip* arguments.

  See the "Configuring Dynamic NAT or PAT" section on page 19-25 for information about the other options.

For example, the following command uses static identity NAT for an inside IP address (10.1.1.3) when accessed by the outside:

```
hostname(config)# static (inside,outside) 10.1.1.3 10.1.1.3 netmask 255.255.255.255
```

The following command uses static identity NAT for an outside address (209.165.201.15) when accessed by the inside:

```
hostname(config)# static (outside,inside) 209.165.201.15 209.165.201.15 netmask
255.255.255.255
```

The following command statically maps an entire subnet:

```
hostname(config)# static (inside,dmz) 10.1.2.0 10.1.2.0 netmask 255.255.255.0
```

The following static identity policy NAT example shows a single real address that uses identity NAT when accessing one destination address, and a translation when accessing another:

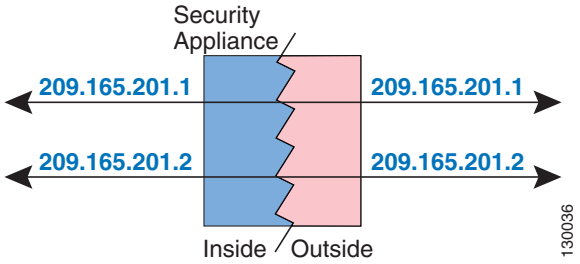
```
hostname(config)# access-list NET1 permit ip host 10.1.2.27 209.165.201.0 255.255.255.224
hostname(config)# access-list NET2 permit ip host 10.1.2.27 209.165.200.224
255.255.255.224
hostname(config)# static (inside,outside) 10.1.2.27 access-list NET1
hostname(config)# static (inside,outside) 209.165.202.130 access-list NET2
```

Configuring NAT Exemption

NAT exemption exempts addresses from translation and allows both real and remote hosts to originate connections. NAT exemption lets you specify the real and destination addresses when determining the real traffic to exempt (similar to policy NAT), so you have greater control using NAT exemption than identity NAT. However unlike policy NAT, NAT exemption does not consider the ports in the access list. Use static identity NAT to consider ports in the access list.

Figure 19-27 shows a typical NAT exemption scenario.

Figure 19-27 NAT Exemption



Note If you remove a NAT exemption configuration, existing connections that use NAT exemption are not affected. To remove these connections, enter the **clear local-host** command.

To configure NAT exemption, enter the following command:

```
hostname(config)# nat (real_interface) 0 access-list acl_name [outside]
```

Create the extended access list using the **access-list extended** command (see the "Adding an Extended Access List" section on page 18-5). This access list can include both **permit** ACEs and **deny** ACEs. Do not specify the real and destination ports in the access list; NAT exemption does not consider the ports. NAT exemption considers the **inactive** and **time-range** keywords, but it does not support ACL with all **inactive** and **time-range** ACEs.

By default, this command exempts traffic from inside to outside. If you want traffic from outside to inside to bypass NAT, then add an additional **nat** command and enter **outside** to identify the NAT instance as outside NAT. You might want to use outside NAT exemption if you configure dynamic NAT for the outside interface and want to exempt other traffic.

For example, to exempt an inside network when accessing any destination address, enter the following command:

```
hostname(config)# access-list EXEMPT permit ip 10.1.2.0 255.255.255.0 any
hostname(config)# nat (inside) 0 access-list EXEMPT
```

To use dynamic outside NAT for a DMZ network, and exempt another DMZ network, enter the following command:

```
hostname(config)# nat (dmz) 1 10.1.2.0 255.255.255.0 outside dns
hostname(config)# global (inside) 1 10.1.1.45
hostname(config)# access-list EXEMPT permit ip 10.1.3.0 255.255.255.0 any
hostname(config)# nat (dmz) 0 access-list EXEMPT
```

To exempt an inside address when accessing two different destination addresses, enter the following commands:

```
hostname(config)# access-list NET1 permit ip 10.1.2.0 255.255.255.0 209.165.201.0
255.255.255.224
hostname(config)# access-list NET1 permit ip 10.1.2.0 255.255.255.0 209.165.200.224
255.255.255.224
hostname(config)# nat (inside) 0 access-list NET1
```

NAT Examples

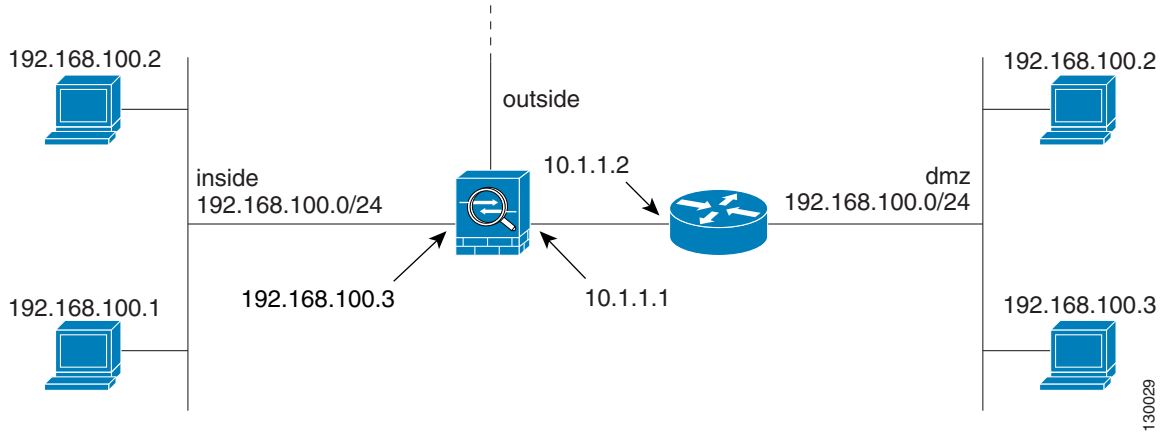
This section describes typical scenarios that use NAT solutions, and includes the following topics:

- Overlapping Networks, page 19-36
- Redirecting Ports, page 19-38

Overlapping Networks

In Figure 19-28, the security appliance connects two private networks with overlapping address ranges.

Figure 19-28 Using Outside NAT with Overlapping Networks



Two networks use an overlapping address space (192.168.100.0/24), but hosts on each network must communicate (as allowed by access lists). Without NAT, when a host on the inside network tries to access a host on the overlapping DMZ network, the packet never makes it past the security appliance, which sees the packet as having a destination address on the inside network. Moreover, if the destination address is being used by another host on the inside network, that host receives the packet.

To solve this problem, use NAT to provide non-overlapping addresses. If you want to allow access in both directions, use static NAT for both networks. If you only want to allow the inside interface to access hosts on the DMZ, then you can use dynamic NAT for the inside addresses, and static NAT for the DMZ addresses you want to access. This example shows static NAT.

To configure static NAT for these two interfaces, perform the following steps. The 10.1.1.0/24 network on the DMZ is not translated.

Step 1 Translate 192.168.100.0/24 on the inside to 10.1.2.0/24 when it accesses the DMZ by entering the following command:

```
hostname(config)# static (inside,dmz) 10.1.2.0 192.168.100.0 netmask 255.255.255.0
```

Step 2 Translate the 192.168.100.0/24 network on the DMZ to 10.1.3.0/24 when it accesses the inside by entering the following command:

```
hostname(config)# static (dmz,inside) 10.1.3.0 192.168.100.0 netmask 255.255.255.0
```

Step 3 Configure the following static routes so that traffic to the dmz network can be routed correctly by the security appliance:

```
hostname(config)# route dmz 192.168.100.128 255.255.255.128 10.1.1.2 1
hostname(config)# route dmz 192.168.100.0 255.255.255.128 10.1.1.2 1
```

The security appliance already has a connected route for the inside network. These static routes allow the security appliance to send traffic for the 192.168.100.0/24 network out the DMZ interface to the gateway router at 10.1.1.2. (You need to split the network into two because you cannot create a static route with the exact same network as a connected route.) Alternatively, you could use a more broad route for the DMZ traffic, such as a default route.

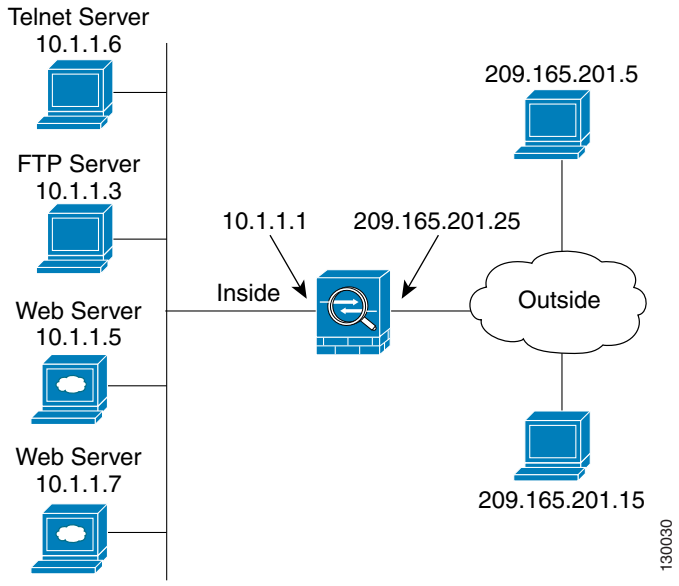
If host 192.168.100.2 on the DMZ network wants to initiate a connection to host 192.168.100.2 on the inside network, the following events occur:

1. The DMZ host 192.168.100.2 sends the packet to IP address 10.1.2.2.
2. When the security appliance receives this packet, the security appliance translates the source address from 192.168.100.2 to 10.1.3.2.
3. Then the security appliance translates the destination address from 10.1.2.2 to 192.168.100.2, and the packet is forwarded.

Redirecting Ports

Figure 19-29 shows an example of a network configuration in which the port redirection feature might be useful.

Figure 19-29 Port Redirection Using Static PAT



In the configuration described in this section, port redirection occurs for hosts on external networks as follows:

- Telnet requests to IP address 209.165.201.5 are redirected to 10.1.1.6.
- FTP requests to IP address 209.165.201.5 are redirected to 10.1.1.3.
- HTTP request to an security appliance outside IP address 209.165.201.25 are redirected to 10.1.1.5.
- HTTP port 8080 requests to PAT address 209.165.201.15 are redirected to 10.1.1.7 port 80.

To implement this configuration, perform the following steps:

Step 1 Configure PAT for the inside network by entering the following commands:

```
hostname(config)# nat (inside) 1 0.0.0.0 0.0.0.0 0 0
hostname(config)# global (outside) 1 209.165.201.15
```

Step 2 Redirect Telnet requests for 209.165.201.5 to 10.1.1.6 by entering the following command:

```
hostname(config)# static (inside,outside) tcp 209.165.201.5 telnet 10.1.1.6 telnet netmask
255.255.255.255
```

Step 3 Redirect FTP requests for IP address 209.165.201.5 to 10.1.1.3 by entering the following command:

```
hostname(config)# static (inside,outside) tcp 209.165.201.5 ftp 10.1.1.3 ftp netmask
255.255.255.255
```

Step 4 Redirect HTTP requests for the security appliance outside interface address to 10.1.1.5 by entering the following command:

```
hostname(config)# static (inside,outside) tcp interface www 10.1.1.5 www netmask
255.255.255.255
```

Step 5 Redirect HTTP requests on port 8080 for PAT address 209.165.201.15 to 10.1.1.7 port 80 by entering the following command:

```
hostname(config)# static (inside,outside) tcp 209.165.201.15 8080 10.1.1.7 www netmask
255.255.255.255
```



CHAPTER 20

Permitting or Denying Network Access

This chapter describes how to control network access through the security appliance using access lists. To create an extended access lists or an EtherType access list, see Chapter 18, "Identifying Traffic with Access Lists."

Note You use ACLs to control network access in both routed and transparent firewall modes. In transparent mode, you can use both extended ACLs (for Layer 3 traffic) and EtherType ACLs (for Layer 2 traffic).

To access the security appliance interface for management access, you do not need an access list allowing the host IP address. You only need to configure management access according to Chapter 42, "Managing System Access."

This chapter includes the following sections:

- Inbound and Outbound Access List Overview, page 20-1
- Applying an Access List to an Interface, page 20-2

Inbound and Outbound Access List Overview

By default, all traffic from a higher-security interface to a lower-security interface is allowed. Access lists let you either allow traffic from lower-security interfaces, or restrict traffic from higher-security interfaces.

The security appliance supports two types of access lists:

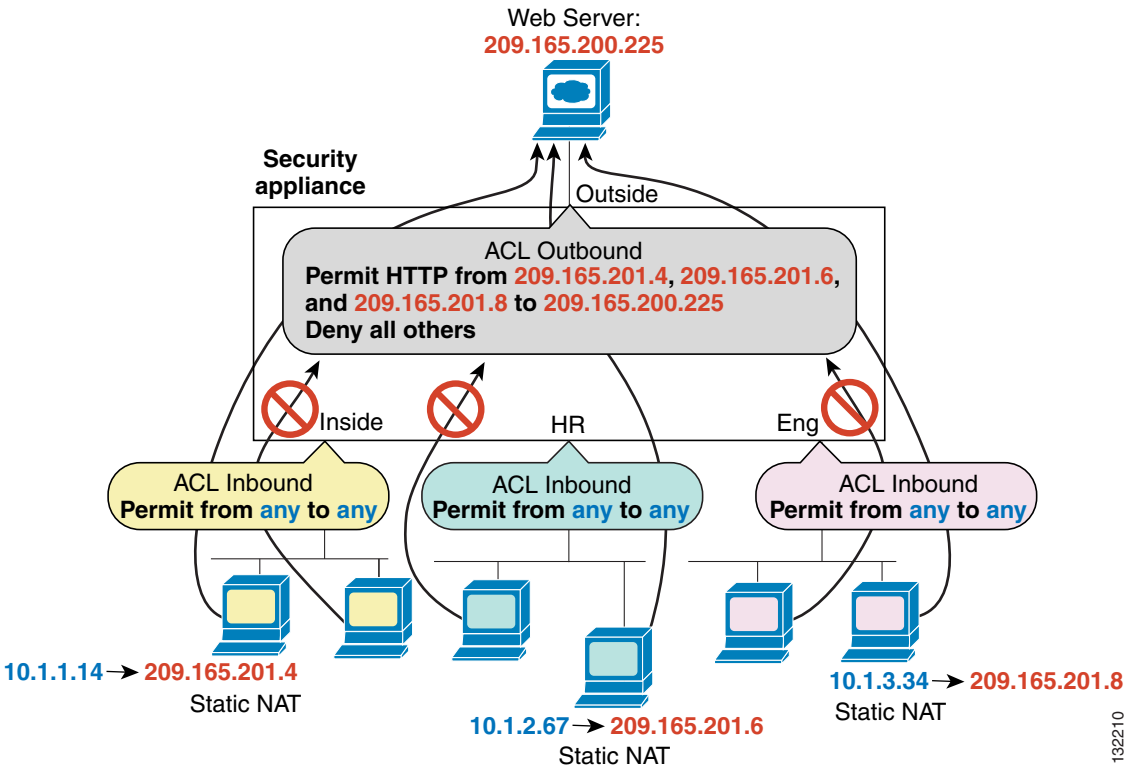
- Inbound—Inbound access lists apply to traffic as it enters an interface.
- Outbound—Outbound access lists apply to traffic as it exits an interface.

Note "Inbound" and "outbound" refer to the application of an access list on an interface, either to traffic entering the security appliance on an interface or traffic exiting the security appliance on an interface. These terms do not refer to the movement of traffic from a lower security interface to a higher security interface, commonly known as inbound, or from a higher to lower interface, commonly known as outbound.

An outbound access list is useful, for example, if you want to allow only certain hosts on the inside networks to access a web server on the outside network. Rather than creating multiple inbound access lists to restrict access, you can create a single outbound access list that allows only the specified hosts

(see Figure 20-1). See the "IP Addresses Used for Access Lists When You Use NAT" section on page 18-3 for information about NAT and IP addresses. The outbound access list prevents any other hosts from reaching the outside network.

Figure 20-1 Outbound Access List



See the following commands for this example:

```
hostname(config)# access-list OUTSIDE extended permit tcp host 209.165.201.4
host 209.165.200.225 eq www
hostname(config)# access-list OUTSIDE extended permit tcp host 209.165.201.6
host 209.165.200.225 eq www
hostname(config)# access-list OUTSIDE extended permit tcp host 209.165.201.8
host 209.165.200.225 eq www
hostname(config)# access-group OUTSIDE out interface outside
```

Applying an Access List to an Interface

To apply an extended access list to the inbound or outbound direction of an interface, enter the following command:

```
hostname(config)# access-group access_list_name {in | out} interface interface_name
[per-user-override]
```

You can apply one access list of each type (extended and EtherType) to both directions of the interface. You can also apply an IPv4 and an IPv6 ACL to an interface at the same time and in the same direction. See the "Inbound and Outbound Access List Overview" section on page 20-1 for more information about access list directions.

The **per-user-override** keyword allows dynamic access lists that are downloaded for user authorization to override the access list assigned to the interface. For example, if the interface access list denies all traffic from 10.0.0.0, but the dynamic access list permits all traffic from 10.0.0.0, then the dynamic access list overrides the interface access list for that user. See the "Configuring RADIUS Authorization" section for more information about per-user access lists. The **per-user-override** keyword is only available for inbound access lists.

For connectionless protocols, you need to apply the access list to the source and destination interfaces if you want traffic to pass in both directions.

The following example illustrates the commands required to enable access to an inside web server with the IP address 209.165.201.12 (this IP address is the address visible on the outside interface after NAT):

```
hostname(config)# access-list ACL_OUT extended permit tcp any host 209.165.201.12 eq www
hostname(config)# access-group ACL_OUT in interface outside
```

You also need to configure NAT for the web server.

The following access lists allow *any* hosts to communicate between the **inside** and **hr** networks, but only specific hosts (209.168.200.3 and 209.168.200.4) to access the outside network, as shown in the last line below:

```
hostname(config)# access-list ANY extended permit ip any any
hostname(config)# access-list OUT extended permit ip host 209.168.200.3 any
hostname(config)# access-list OUT extended permit ip host 209.168.200.4 any

hostname(config)# access-group ANY in interface inside
hostname(config)# access-group ANY in interface hr
hostname(config)# access-group OUT out interface outside
```

For example, the following sample access list allows common EtherTypes originating on the inside interface:

```
hostname(config)# access-list ETHER ethertype permit ipx
hostname(config)# access-list ETHER ethertype permit bpdu
hostname(config)# access-list ETHER ethertype permit mpls-unicast
hostname(config)# access-group ETHER in interface inside
```

The following access list allows some EtherTypes through the security appliance, but denies all others:

```
hostname(config)# access-list ETHER ethertype permit 0x1234
hostname(config)# access-list ETHER ethertype permit bpdu
hostname(config)# access-list ETHER ethertype permit mpls-unicast
hostname(config)# access-group ETHER in interface inside
hostname(config)# access-group ETHER in interface outside
```

The following access list denies traffic with EtherType 0x1256 but allows all others on both interfaces:

```
hostname(config)# access-list nonIP ethertype deny 1256
hostname(config)# access-list nonIP ethertype permit any
hostname(config)# access-group ETHER in interface inside
hostname(config)# access-group ETHER in interface outside
```

The following example uses object groups to permit specific traffic on the inside interface:

```
!
hostname (config)# object-group service myaclog
hostname (config-service)# service-object tcp source range 2000 3000
hostname (config-service)# service-object tcp source range 3000 3010 destinatio$
hostname (config-service)# service-object ipsec
hostname (config-service)# service-object udp destination range 1002 1006
hostname (config-service)# service-object icmp echo
```

```
hostname(config)# access-list outsideacl extended permit object-group myaclog interface
inside any
```



CHAPTER 21

Applying AAA for Network Access

This chapter describes how to enable AAA (pronounced “triple A”) for network access.

For information about AAA for management access, see the “Configuring AAA for System Administrators” section on page 42-5.

This chapter includes the following sections:

- AAA Performance, page 21-1
- Configuring Authentication for Network Access, page 21-1
- Configuring Authorization for Network Access, page 21-8
- Configuring Accounting for Network Access, page 21-14
- Using MAC Addresses to Exempt Traffic from Authentication and Authorization, page 21-16

AAA Performance

The security appliance uses “cut-through proxy” to significantly improve performance compared to a traditional proxy server. The performance of a traditional proxy server suffers because it analyzes every packet at the application layer of the OSI model. The security appliance cut-through proxy challenges a user initially at the application layer and then authenticates against standard AAA servers or the local database. After the security appliance authenticates the user, it shifts the session flow, and all traffic flows directly and quickly between the source and destination while maintaining session state information.

Configuring Authentication for Network Access

This section includes the following topics:

- Authentication Overview, page 21-2
- Enabling Network Access Authentication, page 21-3
- Enabling Secure Authentication of Web Clients, page 21-5
- Authenticating Directly with the Security Appliance, page 21-6

Authentication Overview

The security appliance lets you configure network access authentication using AAA servers. This section includes the following topics:

- One-Time Authentication, page 21-2
- Applications Required to Receive an Authentication Challenge, page 21-2
- Security Appliance Authentication Prompts, page 21-2
- Static PAT and HTTP, page 21-3
- Enabling Network Access Authentication, page 21-3

One-Time Authentication

A user at a given IP address only needs to authenticate one time for all rules and types, until the authentication session expires. (See the **timeout uauth** command in the *Cisco Security Appliance Command Reference* for timeout values.) For example, if you configure the security appliance to authenticate Telnet and FTP, and a user first successfully authenticates for Telnet, then as long as the authentication session exists, the user does not also have to authenticate for FTP.

Applications Required to Receive an Authentication Challenge

Although you can configure the security appliance to require authentication for network access to any protocol or service, users can authenticate directly with HTTP, HTTPS, Telnet, or FTP only. A user must first authenticate with one of these services before the security appliance allows other traffic requiring authentication.

The authentication ports that the security appliance supports for AAA are fixed:

- Port 21 for FTP
- Port 23 for Telnet
- Port 80 for HTTP
- Port 443 for HTTPS

Security Appliance Authentication Prompts

For Telnet and FTP, the security appliance generates an authentication prompt.

For HTTP, the security appliance uses basic HTTP authentication by default, and provides an authentication prompt. You can optionally configure the security appliance to redirect users to an internal web page where they can enter their username and password (configured with the **aaa authentication listener** command).

For HTTPS, the security appliance generates a custom login screen. You can optionally configure the security appliance to redirect users to an internal web page where they can enter their username and password (configured with the **aaa authentication listener** command).

Redirection is an improvement over the basic method because it provides an improved user experience when authenticating, and an identical user experience for HTTP and HTTPS in both Easy VPN and firewall modes. It also supports authenticating directly with the security appliance.

You might want to continue to use basic HTTP authentication if: you do not want the security appliance to open listening ports; if you use NAT on a router and you do not want to create a translation rule for the web page served by the security appliance; basic HTTP authentication might work better with your network. For example non-browser applications, like when a URL is embedded in email, might be more compatible with basic authentication.

After you authenticate correctly, the security appliance redirects you to your original destination. If the destination server also has its own authentication, the user enters another username and password. If you use basic HTTP authentication and need to enter another username and password for the destination server, then you need to configure the **virtual http** command.

Note If you use HTTP authentication, by default the username and password are sent from the client to the security appliance in clear text; in addition, the username and password are sent on to the destination web server as well. See the "Enabling Secure Authentication of Web Clients" section on page 21-5 for information to secure your credentials.

For FTP, a user has the option of entering the security appliance username followed by an at sign (@) and then the FTP username (name1@name2). For the password, the user enters the security appliance password followed by an at sign (@) and then the FTP password (password1@password2). For example, enter the following text.

```
name> jamiec@patm
password> letmein@he110
```

This feature is useful when you have cascaded firewalls that require multiple logins. You can separate several names and passwords by multiple at signs (@).

Static PAT and HTTP

For HTTP authentication, the security appliance checks real ports when static PAT is configured. If it detects traffic destined for real port 80, regardless of the mapped port, the security appliance intercepts the HTTP connection and enforces authentication.

For example, assume that outside TCP port 889 is translated to port 80 (www) and that any relevant access lists permit the traffic:

```
static (inside,outside) tcp 10.48.66.155 889 192.168.123.10 www netmask 255.255.255.255
```

Then when users try to access 10.48.66.155 on port 889, the security appliance intercepts the traffic and enforces HTTP authentication. Users see the HTTP authentication page in their web browsers before the security appliance allows HTTP connection to complete.

If the local port is different than port 80, as in the following example:

```
static (inside,outside) tcp 10.48.66.155 889 192.168.123.10 111 netmask 255.255.255.255
```

Then users do not see the authentication page. Instead, the security appliance sends to the web browser an error message indicating that the user must be authenticated prior using the requested service.

Enabling Network Access Authentication

To enable network access authentication, perform the following steps:

Step 1 Using the **aaa-server** command, identify your AAA servers. If you have already identified your AAA servers, continue to the next step.

For more information about identifying AAA servers, see the "Identifying AAA Server Groups and Servers" section on page 14-9.

Step 2 Using the **access-list** command, create an access list that identifies the source addresses and destination addresses of traffic you want to authenticate. For steps, see the "Adding an Extended Access List" section on page 18-5.

The **permit** ACEs mark matching traffic for authentication, while **deny** entries exclude matching traffic from authentication. Be sure to include the destination ports for either HTTP, HTTPS, Telnet, or FTP in the access list because the user must authenticate with one of these services before other services are allowed through the security appliance.

Step 3 To configure authentication, enter the following command:

```
hostname(config)# aaa authentication match acl_name interface_name server_group
```

Where *acl_name* is the name of the access list you created in Step 2, *interface_name* is the name of the interface as specified with the **nameif** command, and *server_group* is the AAA server group you created in Step 1.

Note You can alternatively use the **aaa authentication include** command (which identifies traffic within the command). However, you cannot use both methods in the same configuration. See the *Cisco Security Appliance Command Reference* for more information.

Step 4 (Optional) To enable the redirection method of authentication for HTTP or HTTPS connections, enter the following command:

```
hostname(config)# aaa authentication listener http[s] interface_name [port portnum]
redirect
```

where the *interface_name* argument is the interface on which you want to enable listening ports.

The **port** *portnum* argument specifies the port number that the security appliance listens on; the defaults are 80 (HTTP) and 443 (HTTPS). You can use any port number and retain the same functionality, but be sure your direct authentication users know the port number; redirected traffic is sent to the correct port number automatically, but direct authenticators must specify the port number manually.

Enter this command separately for HTTP and for HTTPS.

Step 5 (Optional) If you are using the local database for network access authentication and you want to limit the number of consecutive failed login attempts that the security appliance allows any given user account (with the exception of users with a privilege level of 15; this feature does not affect level 15 users), use the following command:

```
hostname(config)# aaa local authentication attempts max-fail number
```

Where number is between 1 and 16.

For example:

```
hostname(config)# aaa local authentication attempts max-fail 7
```

Tip To clear the lockout status of a specific user or all users, use the **clear aaa local user lockout** command.

For example, the following commands authenticate all inside HTTP traffic and SMTP traffic:

```
hostname(config)# aaa-server AuthOutbound protocol tacacs+
hostname(config-aaa-server-group)# exit
hostname(config)# aaa-server AuthOutbound (inside) host 10.1.1.1
hostname(config-aaa-server-host)# key TACPlusUauthKey
hostname(config-aaa-server-host)# exit
hostname(config)# access-list MAIL_AUTH extended permit tcp any any eq smtp
hostname(config)# access-list MAIL_AUTH extended permit tcp any any eq www
hostname(config)# aaa authentication match MAIL_AUTH inside AuthOutbound
hostname(config)# aaa authentication listener http inside redirect
```

The following commands authenticate Telnet traffic from the outside interface to a particular server (209.165.201.5):

```
hostname(config)# aaa-server AuthInbound protocol tacacs+
hostname(config-aaa-server-group)# exit
hostname(config)# aaa-server AuthInbound (inside) host 10.1.1.1
hostname(config-aaa-server-host)# key TACPlusUauthKey
hostname(config-aaa-server-host)# exit
hostname(config)# access-list TELNET_AUTH extended permit tcp any host 209.165.201.5 eq
telnet
hostname(config)# aaa authentication match TELNET_AUTH outside AuthInbound
```

Enabling Secure Authentication of Web Clients

If you use HTTP authentication, by default the username and password are sent from the client to the security appliance in clear text; in addition, the username and password are sent on to the destination web server as well. The security appliance provides several methods of securing HTTP authentication:

- Enable the redirection method of authentication for HTTP—Use the **aaa authentication listener** command with the **redirect** keyword. This method prevents the authentication credentials from continuing to the destination server. See the "Security Appliance Authentication Prompts" section on page 21-2 for more information about the redirection method versus the basic method.
- Enable virtual HTTP—Use the **virtual http** command to let you authenticate separately with the security appliance and with the HTTP server. Even if the HTTP server does not need a second authentication, this command achieves the effect of stripping the basic authentication credentials from the HTTP GET request.
- Enable the exchange of usernames and passwords between a web client and the security appliance with HTTPS—Use the **aaa authentication secure-http-client** command to enable the exchange of usernames and passwords between a web client and the security appliance with HTTPS. This is the only method that protects credentials between the client and the security appliance, as well as between the security appliance and the destination server. You can use this method alone, or in conjunction with either of the other methods so you can maximize your security.

  After enabling this feature, when a user requires authentication when using HTTP, the security appliance redirects the HTTP user to an HTTPS prompt. After you authenticate correctly, the security appliance redirects you to the original HTTP URL.

  Secured web-client authentication has the following limitations:

  – A maximum of 16 concurrent HTTPS authentication sessions are allowed. If all 16 HTTPS authentication processes are running, a new connection requiring authentication will not succeed.

- When **uauth timeout 0** is configured (the **uauth timeout** is set to 0), HTTPS authentication might not work. If a browser initiates multiple TCP connections to load a web page after HTTPS authentication, the first connection is let through, but the subsequent connections trigger authentication. As a result, users are continuously presented with an authentication page, even if the correct username and password are entered each time. To work around this, set the **uauth timeout** to 1 second with the **timeout uauth 0:0:1** command. However, this workaround opens a 1-second window of opportunity that might allow non-authenticated users to go through the firewall if they are coming from the same source IP address.
- Because HTTPS authentication occurs on the SSL port 443, users must not configure an **access-list** command statement to block traffic from the HTTP client to HTTP server on port 443. Furthermore, if static PAT is configured for web traffic on port 80, it must also be configured for the SSL port. In the following example, the first line configures static PAT for web traffic and the second line must be added to support the HTTPS authentication configuration.

```
static (inside,outside) tcp 10.132.16.200 www 10.130.16.10 www
static (inside,outside) tcp 10.132.16.200 443 10.130.16.10 443
```

Authenticating Directly with the Security Appliance

If you do not want to allow HTTP, HTTPS, Telnet, or FTP through the security appliance but want to authenticate other types of traffic, you can authenticate with the security appliance directly using HTTP, HTTPS, or Telnet.

This section includes the following topics:

- Enabling Direct Authentication Using HTTP and HTTPS, page 21-6
- Enabling Direct Authentication Using Telnet, page 21-7

Enabling Direct Authentication Using HTTP and HTTPS

If you enabled the redirect method of HTTP and HTTPS authentication in the "Enabling Network Access Authentication" section on page 21-3, then you also automatically enabled direct authentication.

If you want to continue to use basic HTTP authentication, but want to enable direct authentication for HTTP and HTTPS, then enter the following command:

```
hostname(config)# aaa authentication listener http[s] interface_name [port portnum]
```

where the *interface_name* argument is the interface on which you want to enable direct authentication.

The **port** *portnum* argument specifies the port number that the security appliance listens on; the defaults are 80 (HTTP) and 443 (HTTPS).

Enter this command separately for HTTP and for HTTPS.

If the destination HTTP server requires authentication in addition to the security appliance, then the **virtual http** command lets you authenticate separately with the security appliance (via a AAA server) and with the HTTP server. Without virtual HTTP, the same username and password you used to authenticate with the security appliance is sent to the HTTP server; you are not prompted separately for the HTTP server username and password. Assuming the username and password is not the same for the AAA and HTTP servers, then the HTTP authentication fails.

This command redirects all HTTP connections that require AAA authentication to the virtual HTTP server on the security appliance. The security appliance prompts for the AAA server username and password. After the AAA server authenticates the user, the security appliance redirects the HTTP connection back to the original server, but it does not include the AAA server username and password. Because the username and password are not included in the HTTP packet, the HTTP server prompts the user separately for the HTTP server username and password.

For inbound users (from lower security to higher security), you must also include the virtual HTTP address as a destination interface in the access list applied to the source interface. Moreover, you must add a **static** command for the virtual HTTP IP address, even if NAT is not required (using the **no nat-control** command). An identity NAT command is typically used (where you translate the address to itself).

For outbound users, there is an explicit permit for traffic, but if you apply an access list to an inside interface, be sure to allow access to the virtual HTTP address. A **static** statement is not required.



Note Do not set the **timeout uauth** command duration to 0 seconds when using the **virtual http** command, because this setting prevents HTTP connections to the real web server.

You can authenticate directly with the security appliance at the following URLs when you enable AAA for the interface:

http://*interface_ip*[**:***port*]**/netaccess/connstatus.html**
https://*interface_ip*[**:***port*]**/netaccess/connstatus.html**

Enabling Direct Authentication Using Telnet

Although you can configure network access authentication for any protocol or service (see the **aaa authentication match** or **aaa authentication include** command), you can authenticate directly with HTTP, Telnet, or FTP only. A user must first authenticate with one of these services before other traffic that requires authentication is allowed through. If you do not want to allow HTTP, Telnet, or FTP through the security appliance, but want to authenticate other types of traffic, you can configure virtual Telnet; the user Telnets to a given IP address configured on the security appliance, and the security appliance provides a Telnet prompt.

To configure a virtual Telnet server, enter the following command:

```
hostname(config)# virtual telnet ip_address
```

where the *ip_address* argument sets the IP address for the virtual Telnet server. Make sure this address is an unused address that is routed to the security appliance.

You must configure authentication for Telnet access to the virtual Telnet address as well as the other services you want to authenticate using the **authentication match** or **aaa authentication include** command.

When an unauthenticated user connects to the virtual Telnet IP address, the user is challenged for a username and password, and then authenticated by the AAA server. Once authenticated, the user sees the message “Authentication Successful.” Then, the user can successfully access other services that require authentication.

For inbound users (from lower security to higher security), you must also include the virtual Telnet address as a destination interface in the access list applied to the source interface. Moreover, you must add a **static** command for the virtual Telnet IP address, even if NAT is not required (using the **no nat-control** command). An identity NAT command is typically used (where you translate the address to itself).

For outbound users, there is an explicit permit for traffic, but if you apply an access list to an inside interface, be sure to allow access to the virtual Telnet address. A **static** statement is not required.

To logout from the security appliance, reconnect to the virtual Telnet IP address; you are prompted to log out.

This example shows how to enable virtual Telnet along with AAA authentication for other services:

```
hostname(config)# virtual telnet 209.165.202.129
hostname(config)# access-list ACL-IN extended permit tcp any host 209.165.200.225 eq smtp
hostname(config)# access-list ACL-IN remark This is the SMTP server on the inside
hostname(config)# access-list ACL-IN extended permit tcp any host 209.165.202.129 eq
telnet
hostname(config)# access-list ACL-IN remark This is the virtual Telnet address
hostname(config)# access-group ACL-IN in interface outside
hostname(config)# static (inside, outside) 209.165.202.129 209.165.202.129 netmask
255.255.255.255
hostname(config)# access-list AUTH extended permit tcp any host 209.165.200.225 eq smtp
hostname(config)# access-list AUTH remark This is the SMTP server on the inside
hostname(config)# access-list AUTH extended permit tcp any host 209.165.202.129 eq telnet
hostname(config)# access-list AUTH remark This is the virtual Telnet address
hostname(config)# aaa authentication match AUTH outside tacacs+
```

Configuring Authorization for Network Access

After a user authenticates for a given connection, the security appliance can use authorization to further control traffic from the user.

This section includes the following topics:

- Configuring TACACS+ Authorization, page 21-8
- Configuring RADIUS Authorization, page 21-10

Configuring TACACS+ Authorization

You can configure the security appliance to perform network access authorization with TACACS+. You identify the traffic to be authorized by specifying access lists that authorization rules must match. Alternatively, you can identify the traffic directly in authorization rules themselves.



Tip Using access lists to identify traffic to be authorized can greatly reduced the number of authorization commands you must enter. This is because each authorization rule you enter can specify only one source and destination subnet and service, whereas an access list can include many entries.

Authentication and authorization statements are independent; however, any unauthenticated traffic matched by an authorization statement will be denied. For authorization to succeed, a user must first authenticate with the security appliance. Because a user at a given IP address only needs to authenticate one time for all rules and types, if the authentication session hasn't expired, authorization can occur even if the traffic is matched by an authentication statement.

After a user authenticates, the security appliance checks the authorization rules for matching traffic. If the traffic matches the authorization statement, the security appliance sends the username to the TACACS+ server. The TACACS+ server responds to the security appliance with a permit or a deny for that traffic, based on the user profile. The security appliance enforces the authorization rule in the response.

See the documentation for your TACACS+ server for information about configuring network access authorizations for a user.

To configure TACACS+ authorization, perform the following steps:

Step 1 Enable authentication. For more information, see the "Enabling Network Access Authentication" section on page 21-3. If you have already enabled authentication, continue to the next step.

Step 2 Using the **access-list** command, create an access list that identifies the source addresses and destination addresses of traffic you want to authorize. For steps, see the "Adding an Extended Access List" section on page 18-5.

The **permit** ACEs mark matching traffic for authorization, while **deny** entries exclude matching traffic from authorization. The access list you use for authorization matching should contain rules that are equal to or a subset of the rules in the access list used for authentication matching.



Note If you have configured authentication and want to authorize all the traffic being authenticated, you can use the same access list you created for use with the **aaa authentication match** command.

Step 3 To enable authorization, enter the following command:

```
hostname(config)# aaa authorization match acl_name interface_name server_group
```

where *acl_name* is the name of the access list you created in Step 2, *interface_name* is the name of the interface as specified with the **nameif** command or by default, and *server_group* is the AAA server group you created when you enabled authentication.



Note Alternatively, you can use the **aaa authorization include** command (which identifies traffic within the command) but you cannot use both methods in the same configuration. See the *Cisco Security Appliance Command Reference* for more information.

The following commands authenticate and authorize inside Telnet traffic. Telnet traffic to servers other than 209.165.201.5 can be authenticated alone, but traffic to 209.165.201.5 requires authorization.

```
hostname(config)# access-list TELNET_AUTH extended permit tcp any any eq telnet
hostname(config)# access-list SERVER_AUTH extended permit tcp any host 209.165.201.5 eq
telnet
hostname(config)# aaa-server AuthOutbound protocol tacacs+
hostname(config-aaa-server-group)# exit
hostname(config)# aaa-server AuthOutbound (inside) host 10.1.1.1
hostname(config-aaa-server-host)# key TACPlusUauthKey
hostname(config-aaa-server-host)# exit
hostname(config)# aaa authentication match TELNET_AUTH inside AuthOutbound
hostname(config)# aaa authorization match SERVER_AUTH inside AuthOutbound
```

Configuring RADIUS Authorization

When authentication succeeds, the RADIUS protocol returns user authorizations in the access-accept message sent by a RADIUS server. For more information about configuring authentication, see the "Configuring Authentication for Network Access" section on page 21-1.

When you configure the security appliance to authenticate users for network access, you are also implicitly enabling RADIUS authorizations; therefore, this section contains no information about configuring RADIUS authorization on the security appliance. It does provide information about how the security appliance handles access list information received from RADIUS servers.

You can configure a RADIUS server to download an access list to the security appliance or an access list name at the time of authentication. The user is authorized to do only what is permitted in the user-specific access list.

Note If you have used the **access-group** command to apply access lists to interfaces, be aware of the following effects of the **per-user-override** keyword on authorization by user-specific access lists:

- Without the **per-user-override** keyword, traffic for a user session must be permitted by both the interface access list and the user-specific access list.
- With the **per-user-override** keyword, the user-specific access list determines what is permitted.

For more information, see the **access-group** command entry in the *Cisco Security Appliance Command Reference*.

This section includes the following topics:

- Configuring a RADIUS Server to Send Downloadable Access Control Lists, page 21-10
- Configuring a RADIUS Server to Download Per-User Access Control List Names, page 21-14

Configuring a RADIUS Server to Send Downloadable Access Control Lists

This section describes how to configure Cisco Secure ACS or a third-party RADIUS server, and includes the following topics:

- About the Downloadable Access List Feature and Cisco Secure ACS, page 21-10
- Configuring Cisco Secure ACS for Downloadable Access Lists, page 21-12
- Configuring Any RADIUS Server for Downloadable Access Lists, page 21-13
- Converting Wildcard Netmask Expressions in Downloadable Access Lists, page 21-14

About the Downloadable Access List Feature and Cisco Secure ACS

Downloadable access lists is the most scalable means of using Cisco Secure ACS to provide the appropriate access lists for each user. It provides the following capabilities:

- Unlimited access list size—Downloadable access lists are sent using as many RADIUS packets as required to transport the full access list from Cisco Secure ACS to the security appliance.
- Simplified and centralized management of access lists—Downloadable access lists enable you to write a set of access lists once and apply it to many user or group profiles and distribute it to many security appliances.

This approach is most useful when you have very large access list sets that you want to apply to more than one Cisco Secure ACS user or group; however, its ability to simplify Cisco Secure ACS user and group management makes it useful for access lists of any size.

The security appliance receives downloadable access lists from Cisco Secure ACS using the following process:

1. The security appliance sends a RADIUS authentication request packet for the user session.
2. If Cisco Secure ACS successfully authenticates the user, Cisco Secure ACS returns a RADIUS access-accept message that contains the internal name of the applicable downloadable access list. The Cisco IOS cisco-av-pair RADIUS VSA (vendor 9, attribute 1) contains the following attribute-value pair to identify the downloadable access list set:

   `ACS:CiscoSecure-Defined-ACL=`*`acl-set-name`*

   where *acl-set-name* is the internal name of the downloadable access list, which is a combination of the name assigned to the access list by the Cisco Secure ACS administrator and the date and time that the access list was last modified.
3. The security appliance examines the name of the downloadable access list and determines if it has previously received the named downloadable access list.
   - If the security appliance has previously received the named downloadable access list, communication with Cisco Secure ACS is complete and the security appliance applies the access list to the user session. Because the name of the downloadable access list includes the date and time it was last modified, matching the name sent by Cisco Secure ACS to the name of an access list previous downloaded means that the security appliance has the most recent version of the downloadable access list.
   - If the security appliance has not previously received the named downloadable access list, it may have an out-of-date version of the access list or it may not have downloaded any version of the access list. In either case, the security appliance issues a RADIUS authentication request using the downloadable access list name as the username in the RADIUS request and a null password attribute. In a cisco-av-pair RADIUS VSA, the request also includes the following attribute-value pairs:

     ```
     AAA:service=ip-admission
     AAA:event=acl-download
     ```

     In addition, the security appliance signs the request with the Message-Authenticator attribute (IETF RADIUS attribute 80).
4. Upon receipt of a RADIUS authentication request that has a username attribute containing the name of a downloadable access list, Cisco Secure ACS authenticates the request by checking the Message-Authenticator attribute. If the Message-Authenticator attribute is missing or incorrect, Cisco Secure ACS ignores the request. The presence of the Message-Authenticator attribute prevents malicious use of a downloadable access list name to gain unauthorized network access. The Message-Authenticator attribute and its use are defined in RFC 2869, RADIUS Extensions, available at http://www.ietf.org.
5. If the access list required is less than approximately 4 KB in length, Cisco Secure ACS responds with an access-accept message containing the access list. The largest access list that can fit in a single access-accept message is slightly less than 4 KB because some of the message must be other required attributes.

   Cisco Secure ACS sends the downloadable access list in a cisco-av-pair RADIUS VSA. The access list is formatted as a series of attribute-value pairs that each contain an ACE and are numbered serially:

   `ip:inacl#1=`*`ACE-1`*

```
ip:inacl#2=ACE-2
.
.
.
ip:inacl#n=ACE-n
```

An example of an attribute-value pair follows:

```
ip:inacl#1=permit tcp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
```

6. If the access list required is more than approximately 4 KB in length, Cisco Secure ACS responds with an access-challenge message that contains a portion of the access list, formatted as described above, and an State attribute (IETF RADIUS attribute 24), which contains control data used by Cisco Secure ACS to track the progress of the download. Cisco Secure ACS fits as many complete attribute-value pairs into the cisco-av-pair RADIUS VSA as it can without exceeding the maximum RADIUS message size.

   The security appliance stores the portion of the access list received and responds with another access-request message containing the same attributes as the first request for the downloadable access list plus a copy of the State attribute received in the access-challenge message.

   This repeats until Cisco Secure ACS sends the last of the access list in an access-accept message.

Configuring Cisco Secure ACS for Downloadable Access Lists

You can configure downloadable access lists on Cisco Secure ACS as a shared profile component and then assign the access list to a group or to an individual user.

The access list definition consists of one or more security appliance commands that are similar to the extended **access-list** command (see the "Adding an Extended Access List" section on page 18-5), except without the following prefix:

access-list *acl_name* **extended**

The following example is a downloadable access list definition on Cisco Secure ACS version 3.3:

```
+-------------------------------------------+
| Shared profile Components                 |
|                                           |
|      Downloadable IP ACLs Content         |
|                                           |
| Name:    acs_ten_acl                      |
|                                           |
|      ACL Definitions                      |
|                                           |
| permit tcp any host 10.0.0.254            |
| permit udp any host 10.0.0.254            |
| permit icmp any host 10.0.0.254           |
| permit tcp any host 10.0.0.253            |
| permit udp any host 10.0.0.253            |
| permit icmp any host 10.0.0.253           |
| permit tcp any host 10.0.0.252            |
| permit udp any host 10.0.0.252            |
| permit icmp any host 10.0.0.252           |
| permit ip any any                         |
+-------------------------------------------+
```

For more information about creating downloadable access lists and associating them with users, see the user guide for your version of Cisco Secure ACS.

On the security appliance, the downloaded access list has the following name:

```
#ACSACL#-ip-acl_name-number
```

The *acl_name* argument is the name that is defined on Cisco Secure ACS (acs_ten_acl in the preceding example), and *number* is a unique version ID generated by Cisco Secure ACS.

The downloaded access list on the security appliance consists of the following lines:

```
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit tcp any host 10.0.0.254
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit udp any host 10.0.0.254
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit icmp any host 10.0.0.254
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit tcp any host 10.0.0.253
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit udp any host 10.0.0.253
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit icmp any host 10.0.0.253
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit tcp any host 10.0.0.252
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit udp any host 10.0.0.252
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit icmp any host 10.0.0.252
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit ip any any
```

Configuring Any RADIUS Server for Downloadable Access Lists

You can configure any RADIUS server that supports Cisco IOS RADIUS VSAs to send user-specific access lists to the security appliance in a Cisco IOS RADIUS cisco-av-pair VSA (vendor 9, attribute 1).

In the cisco-av-pair VSA, configure one or more ACEs that are similar to the **access-list extended** command (see the "Adding an Extended Access List" section on page 18-5), except that you replace the following command prefix:

access-list *acl_name* **extended**

with the following text:

ip:inacl#*nnn***=**

The *nnn* argument is a number in the range from 0 to 999999999 that identifies the order of the command statement to be configured on the security appliance. If this parameter is omitted, the sequence value is 0, and the order of the ACEs inside the cisco-av-pair RADIUS VSA is used.

The following example is an access list definition as it should be configured for a cisco-av-pair VSA on a RADIUS server:

```
ip:inacl#1=permit tcp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
ip:inacl#99=deny tcp any any
ip:inacl#2=permit udp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
ip:inacl#100=deny udp any any
ip:inacl#3=permit icmp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
```

For information about making unique per user the access lists that are sent in the cisco-av-pair attribute, see the documentation for your RADIUS server.

On the security appliance, the downloaded access list name has the following format:

```
AAA-user-username
```

The *username* argument is the name of the user that is being authenticated.

The downloaded access list on the security appliance consists of the following lines. Notice the order based on the numbers identified on the RADIUS server.

```
access-list  AAA-user-bcham34-79AD4A08 permit tcp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
access-list  AAA-user-bcham34-79AD4A08 permit udp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
access-list  AAA-user-bcham34-79AD4A08 permit icmp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
access-list  AAA-user-bcham34-79AD4A08 deny tcp any any
access-list  AAA-user-bcham34-79AD4A08 deny udp any any
```

Downloaded access lists have two spaces between the word “access-list” and the name. These spaces serve to differentiate a downloaded access list from a local access list. In this example, “79AD4A08” is a hash value generated by the security appliance to help determine when access list definitions have changed on the RADIUS server.

Converting Wildcard Netmask Expressions in Downloadable Access Lists

If a RADIUS server provides downloadable access lists to Cisco VPN 3000 series concentrators as well as to the security appliance, you may need the security appliance to convert wildcard netmask expressions to standard netmask expressions. This is because Cisco VPN 3000 series concentrators support wildcard netmask expressions but the security appliance only supports standard netmask expressions. Configuring the security appliance to convert wildcard netmask expressions helps minimize the effects of these differences upon how you configure downloadable access lists on your RADIUS servers. Translation of wildcard netmask expressions means that downloadable access lists written for Cisco VPN 3000 series concentrators can be used by the security appliance without altering the configuration of the downloadable access lists on the RADIUS server.

You configure access list netmask conversion on a per-server basis, using the **acl-netmask-convert** command, available in the aaa-server configuration mode. For more information about configuring a RADIUS server, see “Identifying AAA Server Groups and Servers” section on page 14-9. For more information about the **acl-netmask-convert** command, see the *Cisco Security Appliance Command Reference*.

Configuring a RADIUS Server to Download Per-User Access Control List Names

To download a name for an access list that you already created on the security appliance from the RADIUS server when a user authenticates, configure the IETF RADIUS filter-id attribute (attribute number 11) as follows:

filter-id=*acl_name*

Note In Cisco Secure ACS, the value for filter-id attributes are specified in boxes in the HTML interface, omitting **filter-id=** and entering only *acl_name*.

For information about making unique per user the filter-id attribute value, see the documentation for your RADIUS server.

See the “Adding an Extended Access List” section on page 18-5 to create an access list on the security appliance.

Configuring Accounting for Network Access

The security appliance can send accounting information to a RADIUS or TACACS+ server about any TCP or UDP traffic that passes through the security appliance. If that traffic is also authenticated, then the AAA server can maintain accounting information by username. If the traffic is not authenticated, the AAA server can maintain accounting information by IP address. Accounting information includes when sessions start and stop, username, the number of bytes that pass through the security appliance for the session, the service used, and the duration of each session.

To configure accounting, perform the following steps:

Step 1 If you want the security appliance to provide accounting data per user, you must enable authentication. For more information, see the "Enabling Network Access Authentication" section on page 21-3. If you want the security appliance to provide accounting data per IP address, enabling authentication is not necessary and you can continue to the next step.

Step 2 Using the **access-list** command, create an access list that identifies the source addresses and destination addresses of traffic you want accounted. For steps, see the "Adding an Extended Access List" section on page 18-5.

The **permit** ACEs mark matching traffic for authorization, while **deny** entries exclude matching traffic from authorization.

Note If you have configured authentication and want accounting data for all the traffic being authenticated, you can use the same access list you created for use with the **aaa authentication match** command.

Step 3 To enable accounting, enter the following command:

```
hostname(config)# aaa accounting match acl_name interface_name server_group
```

where the *acl_name* argument is the access list name set in the **access-list** command.

The *interface_name* argument is the interface name set in the **nameif** command.

The *server_group* argument is the server group name set in the **aaa-server** command.

Note Alternatively, you can use the **aaa accounting include** command (which identifies traffic within the command) but you cannot use both methods in the same configuration. See the *Cisco Security Appliance Command Reference* for more information.

The following commands authenticate, authorize, and account for inside Telnet traffic. Telnet traffic to servers other than 209.165.201.5 can be authenticated alone, but traffic to 209.165.201.5 requires authorization and accounting.

```
hostname(config)# aaa-server AuthOutbound protocol tacacs+
hostname(config-aaa-server-group)# exit
hostname(config)# aaa-server AuthOutbound (inside) host 10.1.1.1
hostname(config-aaa-server-host)# key TACPlusUauthKey
hostname(config-aaa-server-host)# exit
hostname(config)# access-list TELNET_AUTH extended permit tcp any any eq telnet
hostname(config)# access-list SERVER_AUTH extended permit tcp any host 209.165.201.5 eq
telnet
hostname(config)# aaa authentication match TELNET_AUTH inside AuthOutbound
hostname(config)# aaa authorization match SERVER_AUTH inside AuthOutbound
hostname(config)# aaa accounting match SERVER_AUTH inside AuthOutbound
```

Using MAC Addresses to Exempt Traffic from Authentication and Authorization

The security appliance can exempt from authentication and authorization any traffic from specific MAC addresses. For example, if the security appliance authenticates TCP traffic originating on a particular network but you want to allow unauthenticated TCP connections from a specific server, you would use a MAC exempt rule to exempt from authentication and authorization any traffic from the server specified by the rule.

This feature is particularly useful to exempt devices such as IP phones that cannot respond to authentication prompts.

To use MAC addresses to exempt traffic from authentication and authorization, perform the following steps:

Step 1 To configure a MAC list, enter the following command:

```
hostname(config)# mac-list id {deny | permit} mac macmask
```

Where the *id* argument is the hexadecimal number that you assign to the MAC list. To group a set of MAC addresses, enter the **mac-list** command as many times as needed with the same ID value. Because you can only use one MAC list for AAA exemption, be sure that your MAC list includes all the MAC addresses you want to exempt. You can create multiple MAC lists, but you can only use one at a time.

The order of entries matters, because the packet uses the first entry it matches, as opposed to a best match scenario. If you have a permit entry, and you want to deny an address that is allowed by the permit entry, be sure to enter the deny entry before the permit entry.

The *mac* argument specifies the source MAC address in 12-digit hexadecimal form; that is, nnnn.nnnn.nnnn.

The *macmask* argument specifies the portion of the MAC address that should be used for matching. For example, ffff.ffff.ffff matches the MAC address exactly. ffff.ffff.0000 matches only the first 8 digits.

Step 2 To exempt traffic for the MAC addresses specified in a particular MAC list, enter the following command:

```
hostname(config)# aaa mac-exempt match id
```

Where *id* is the string identifying the MAC list containing the MAC addresses whose traffic is to be exempt from authentication and authorization. You can only enter one instance of the **aaa mac-exempt** command.

The following example bypasses authentication for a single MAC address:

```
hostname(config)# mac-list abc permit 00a0.c95d.0282 ffff.ffff.ffff
hostname(config)# aaa mac-exempt match abc
```

The following entry bypasses authentication for all Cisco IP Phones, which have the hardware ID 0003.E3:

```
hostname(config)# mac-list acd permit 0003.E300.0000 FFFF.FF00.0000
hostname(config)# aaa mac-exempt match acd
```

The following example bypasses authentication for a a group of MAC addresses except for 00a0.c95d.02b2. Enter the deny statement before the permit statement, because 00a0.c95d.02b2 matches the permit statement as well, and if it is first, the deny statement will never be matched.

```
hostname(config)# mac-list 1 deny 00a0.c95d.0282 ffff.ffff.ffff
hostname(config)# mac-list 1 permit 00a0.c95d.0000 ffff.ffff.0000
hostname(config)# aaa mac-exempt match 1
```



CHAPTER 22

Applying Filtering Services

This chapter describes ways to filter web traffic to reduce security risks or prevent inappropriate use. This chapter includes the following sections:

- Filtering Overview, page 22-1
- Filtering ActiveX Objects, page 22-2
- Filtering Java Applets, page 22-3
- Filtering URLs and FTP Requests with an External Server, page 22-4
- Viewing Filtering Statistics and Configuration, page 22-9

Filtering Overview

This section describes how filtering can provide greater control over traffic passing through the security appliance. Filtering can be used in two distinct ways:

- Filtering ActiveX objects or Java applets
- Filtering with an external filtering server

Instead of blocking access altogether, you can remove specific undesirable objects from HTTP traffic, such as ActiveX objects or Java applets, that may pose a security threat in certain situations.

You can also use URL filtering to direct specific traffic to an external filtering server, such an Secure Computing SmartFilter (formerly N2H2) or Websense filtering server. Long URL, HTTPS, and FTP filtering can now be enabled using both Websense and Secure Computing SmartFilter for URL filtering. Filtering servers can block traffic to specific sites or types of sites, as specified by the security policy.

Note URL caching will only work if the version of the URL server software from the URL server vender supports it.

Because URL filtering is CPU-intensive, using an external filtering server ensures that the throughput of other traffic is not affected. However, depending on the speed of your network and the capacity of your URL filtering server, the time required for the initial connection may be noticeably slower when filtering traffic with an external filtering server.

Filtering ActiveX Objects

This section describes how to apply filtering to remove ActiveX objects from HTTP traffic passing through the firewall. This section includes the following topics:

- ActiveX Filtering Overview, page 22-2
- Enabling ActiveX Filtering, page 22-2

ActiveX Filtering Overview

ActiveX objects may pose security risks because they can contain code intended to attack hosts and servers on a protected network. You can disable ActiveX objects with ActiveX filtering.

ActiveX controls, formerly known as OLE or OCX controls, are components you can insert in a web page or other application. These controls include custom forms, calendars, or any of the extensive third-party forms for gathering or displaying information. As a technology, ActiveX creates many potential problems for network clients including causing workstations to fail, introducing network security problems, or being used to attack servers.

The **filter activex** command blocks the HTML <object> commands by commenting them out within the HTML web page. ActiveX filtering of HTML files is performed by selectively replacing the <APPLET> and </APPLET> and <OBJECT CLASSID> and </OBJECT> tags with comments. Filtering of nested tags is supported by converting top-level tags to comments.

Caution This command also blocks any Java applets, image files, or multimedia objects that are embedded in object tags.

If the <object> or </object> HTML tags split across network packets or if the code in the tags is longer than the number of bytes in the MTU, security appliance cannot block the tag.

ActiveX blocking does not occur when users access an IP address referenced by the **alias** command or for WebVPN traffic.

Enabling ActiveX Filtering

This section describes how to remove ActiveX objects in HTTP traffic passing through the security appliance. To remove ActiveX objects, enter the following command in global configuration mode:

```
hostname(config)# filter activex port[-port] local_ip local_mask foreign_ip foreign_mask
```

To use this command, replace port with the TCP port to which filtering is applied. Typically, this is port 80, but other values are accepted. The **http** or **url** literal can be used for port 80. You can specify a range of ports by using a hyphen between the starting port number and the ending port number.

The local IP address and mask identify one or more internal hosts that are the source of the traffic to be filtered. The foreign address and mask specify the external destination of the traffic to be filtered.

You can set either address to **0.0.0.0** (or in shortened form, **0**) to specify all hosts. You can use **0.0.0.0** for either mask (or in shortened form, **0**) to specify all hosts.

The following example specifies that ActiveX objects are blocked on all outbound connections:

```
hostname(config)# filter activex 80 0 0 0 0
```

This command specifies that the ActiveX object blocking applies to web traffic on port 80 from any local host and for connections to any foreign host.

To remove the configuration, use the **no** form of the command, as in the following example:

```
hostname(config)# no filter activex 80 0 0 0 0
```

Filtering Java Applets

This section describes how to apply filtering to remove Java applets from HTTP traffic passing through the firewall. Java applets may pose security risks because they can contain code intended to attack hosts and servers on a protected network. You can remove Java applets with the **filter java** command.

The **filter java** command filters out Java applets that return to the security appliance from an outbound connection. The user still receives the HTML page, but the web page source for the applet is commented out so that the applet cannot execute. The **filter java** command does not filter WebVPN traffic.

Note Use the **filter activex** command to remove Java applets that are embedded in <object> tags.

To remove Java applets in HTTP traffic passing through the firewall, enter the following command in global configuration mode:

```
hostname(config)# filter java port[-port] local_ip local_mask foreign_ip foreign_mask
```

To use this command, replace port with the TCP port to which filtering is applied. Typically, this is port 80, but other values are accepted. The **http** or **url** literal can be used for port 80. You can specify a range of ports by using a hyphen between the starting port number and the ending port number.

The local IP address and mask identify one or more internal hosts that are the source of the traffic to be filtered. The foreign address and mask specify the external destination of the traffic to be filtered.

You can set either address to **0.0.0.0** (or in shortened form, **0**) to specify all hosts. You can use **0.0.0.0** for either mask (or in shortened form, **0**) to specify all hosts.

You can set either address to **0.0.0.0** (or in shortened form, **0**) to specify all hosts. You can use **0.0.0.0** for either mask (or in shortened form, **0**) to specify all hosts.

The following example specifies that Java applets are blocked on all outbound connections:

```
hostname(config)# filter java 80 0 0 0 0
```

This command specifies that the Java applet blocking applies to web traffic on port 80 from any local host and for connections to any foreign host.

The following example blocks downloading of Java applets to a host on a protected network:

```
hostname(config)# filter java http 192.168.3.3 255.255.255.255 0 0
```

This command prevents host 192.168.3.3 from downloading Java applets.

To remove the configuration, use the **no** form of the command, as in the following example:

```
hostname(config)# no filter java http 192.168.3.3 255.255.255.255 0 0
```

Filtering URLs and FTP Requests with an External Server

This section describes how to filter URLs and FTP requests with an external server. This section includes the following topics:

- URL Filtering Overview, page 22-4
- Identifying the Filtering Server, page 22-4
- Buffering the Content Server Response, page 22-6
- Caching Server Addresses, page 22-6
- Filtering HTTP URLs, page 22-7
- Filtering HTTPS URLs, page 22-8
- Filtering FTP Requests, page 22-9

URL Filtering Overview

You can apply filtering to connection requests originating from a more secure network to a less secure network. Although you can use ACLs to prevent outbound access to specific content servers, managing usage this way is difficult because of the size and dynamic nature of the Internet. You can simplify configuration and improve security appliance performance by using a separate server running one of the following Internet filtering products:

- Websense Enterprise for filtering HTTP, HTTPS, and FTP.
- Secure Computing SmartFilter (formerly N2H2) for filtering HTTP, HTTPS, FTP, and long URL filtering.

Note URL caching will only work if the version of the URL server software from the URL server vender supports it.

Although security appliance performance is less affected when using an external server, users may notice longer access times to websites or FTP servers when the filtering server is remote from the security appliance.

When filtering is enabled and a request for content is directed through the security appliance, the request is sent to the content server and to the filtering server at the same time. If the filtering server allows the connection, the security appliance forwards the response from the content server to the originating client. If the filtering server denies the connection, the security appliance drops the response and sends a message or return code indicating that the connection was not successful.

If user authentication is enabled on the security appliance, then the security appliance also sends the user name to the filtering server. The filtering server can use user-specific filtering settings or provide enhanced reporting regarding usage.

Identifying the Filtering Server

You can identify up to four filtering servers per context. The security appliance uses the servers in order until a server responds. You can only configure a single type of server (Websense or Secure Computing SmartFilter) in your configuration.

Note You must add the filtering server before you can configure filtering for HTTP or HTTPS with the **filter** command. If you remove the filtering servers from the configuration, then all **filter** commands are also removed.

Identify the address of the filtering server using the **url-server** command:

For Websense:

```
hostname(config)# url-server (if_name) host local_ip [timeout seconds] [protocol TCP | UDP
version [1|4] [connections num_conns] ]
```

For Secure Computing SmartFilter (formerly N2H2):

```
hostname(config)# url-server (if_name) vendor {secure-computing | n2h2} host
<local_ip> [port <number>] [timeout <seconds>] [protocol {TCP [connections <number>]} |
UDP]
```

where <*if_name*> is the name of the security appliance interface connected to the filtering server (the default is inside).

For the **vendor** {secure-computing | n2h2}, you can use ‘secure-computing as a vendor string, however, ‘n2h2’ is acceptable for backward compatibility. When the configuration entries are generated, ‘secure-computing’ is saved as the vendor string.

The **host** <*local_ip*> is the IP address of the URL filtering server.

The **port** <*number*> is the Secure Computing SmartFilter server port number of the filtering server; the security appliance also listens for UDP replies on this port.

Note The default port is 4005. This is the default port used by the Secure Computing SmartFilter server to communicate to the security appliance via TCP or UDP. For information on changing the default port, please refer to the *Filtering by N2H2 Administrator's Guide*.

The **timeout** <*seconds*> is the number of seconds the security appliance should keep trying to connect to the filtering server.

The **connections** <*number*> is the number of tries to attempt to make a connection between the host and server.

For example, to identify a single Websense filtering server, enter the following command:

```
hostname(config)# url-server (perimeter) host 10.0.1.1 protocol TCP version 4
```

This identifies a Websense filtering server with the IP address 10.0.1.1 on a perimeter interface of the security appliance.Version 4, which is enabled in this example, is recommended by Websense because it supports caching.

To identify redundant Secure Computing SmartFilter servers, enter the following commands:

```
hostname(config)# url-server (perimeter) vendor n2h2 host 10.0.1.1
hostname(config)# url-server (perimeter) vendor n2h2 host 10.0.1.2
```

This identifies two Sentian filtering servers, both on a perimeter interface of the security appliance.

Buffering the Content Server Response

When a user issues a request to connect to a content server, the security appliance sends the request to the content server and to the filtering server at the same time. If the filtering server does not respond before the content server, the server response is dropped. This delays the web server response from the point of view of the web client because the client must reissue the request.

By enabling the HTTP response buffer, replies from web content servers are buffered and the responses are forwarded to the requesting client if the filtering server allows the connection. This prevents the delay that might otherwise occur.

To configure buffering for responses to HTTP or FTP requests, perform the following steps:

Step 1 To enable buffering of responses for HTTP or FTP requests that are pending a response from the filtering server, enter the following command:

```
hostname(config)# url-block block block-buffer-limit
```

Replace *block-buffer* with the maximum number of HTTP responses that can be buffered while awaiting responses from the url-server.

Note Buffering URLs longer than 3072 bytes are not supported.

Step 2 To configure the maximum memory available for buffering pending URLs (and for buffering long URLs), enter the following command:

```
hostname(config)# url-block mempool-size memory-pool-size
```

Replace memory-pool-size with a value from 2 to 10240 for a maximum memory allocation of 2 KB to 10 MB.

Caching Server Addresses

After a user accesses a site, the filtering server can allow the security appliance to cache the server address for a certain amount of time, as long as every site hosted at the address is in a category that is permitted at all times. Then, when the user accesses the server again, or if another user accesses the server, the security appliance does not need to consult the filtering server again.

Note Requests for cached IP addresses are not passed to the filtering server and are not logged. As a result, this activity does not appear in any reports. You can accumulate Websense run logs before using the **url-cache** command.

Use the **url-cache** command if needed to improve throughput, as follows:

```
hostname(config)# url-cache dst | src_dst size
```

Replace *size* with a value for the cache size within the range 1 to 128 (KB).

Use the **dst** keyword to cache entries based on the URL destination address. Select this mode if all users share the same URL filtering policy on the Websense server.

Use the **src_dst** keyword to cache entries based on both the source address initiating the URL request as well as the URL destination address. Select this mode if users do not share the same URL filtering policy on the Websense server.

Filtering HTTP URLs

This section describes how to configure HTTP filtering with an external filtering server. This section includes the following topics:

- Configuring HTTP Filtering, page 22-7
- Enabling Filtering of Long HTTP URLs, page 22-7
- Truncating Long HTTP URLs, page 22-7
- Exempting Traffic from Filtering, page 22-8

Configuring HTTP Filtering

You must identify and enable the URL filtering server before enabling HTTP filtering.

When the filtering server approves an HTTP connection request, the security appliance allows the reply from the web server to reach the originating client. If the filtering server denies the request, the security appliance redirects the user to a block page, indicating that access was denied.

To enable HTTP filtering, enter the following command:

```
hostname(config)# filter url [http | port[-port] local_ip local_mask foreign_ip
foreign_mask] [allow] [proxy-block]
```

Replace *port* with one or more port numbers if a different port than the default port for HTTP (80) is used. Replace *local_ip* and *local_mask* with the IP address and subnet mask of a user or subnetwork making requests. Replace *foreign_ip* and *foreign_mask* with the IP address and subnet mask of a server or subnetwork responding to requests.

The **allow** option causes the security appliance to forward HTTP traffic without filtering when the primary filtering server is unavailable. Use the **proxy-block** command to drop all requests to proxy servers.

Enabling Filtering of Long HTTP URLs

By default, the security appliance considers an HTTP URL to be a long URL if it is greater than 1159 characters. You can increase the maximum length allowed.

Configure the maximum size of a single URL with the following command:

```
hostname(config)# url-block url-size long-url-size
```

Replace long-url-size with the maximum size in KB for each long URL being buffered. For Websense, this is a value from 2 to 4 for a maximum URL size of 2 KB to 4 KB; for Secure Computing, this is a value between 2 to 3 for a maximum URL size of 2 KB to 3 KB. The default value is 2.

Truncating Long HTTP URLs

By default, if a URL exceeds the maximum permitted size, then it is dropped. To avoid this, you can set the security appliance to truncate a long URL by entering the following command:

```
hostname(config)# filter url [longurl-truncate | longurl-deny | cgi-truncate]
```

The **longurl-truncate** option causes the security appliance to send only the hostname or IP address portion of the URL for evaluation to the filtering server when the URL is longer than the maximum length permitted. Use the **longurl-deny** option to deny outbound URL traffic if the URL is longer than the maximum permitted.

Use the **cgi-truncate** option to truncate CGI URLs to include only the CGI script location and the script name without any parameters. Many long HTTP requests are CGI requests. If the parameters list is very long, waiting and sending the complete CGI request including the parameter list can use up memory resources and affect firewall performance.

Exempting Traffic from Filtering

To exempt specific traffic from filtering, enter the following command:

```
hostname(config)# filter url except source_ip source_mask dest_ip dest_mask
```

For example, the following commands cause all HTTP requests to be forwarded to the filtering server except for those from 10.0.2.54.

```
hostname(config)# filter url http 0 0 0 0
hostname(config)# filter url except 10.0.2.54 255.255.255.255 0 0
```

Filtering HTTPS URLs

You must identify and enable the URL filtering server before enabling HTTPS filtering.

Note Websense and Smartfilter currently support HTTPS; older versions of Secure Computing SmartFilter (formerly N2H2) did not support HTTPS filtering.

Because HTTPS content is encrypted, the security appliance sends the URL lookup without directory and filename information. When the filtering server approves an HTTPS connection request, the security appliance allows the completion of SSL connection negotiation and allows the reply from the web server to reach the originating client. If the filtering server denies the request, the security appliance prevents the completion of SSL connection negotiation. The browser displays an error message such as “The Page or the content cannot be displayed.”

Note The security appliance does not provide an authentication prompt for HTTPS, so a user must authenticate with the security appliance using HTTP or FTP before accessing HTTPS servers.

To enable HTTPS filtering, enter the following command:

```
hostname(config)# filter https port[-port] localIP local_mask foreign_IP foreign_mask
[allow]
```

Replace *port*[*-port*] with a range of port numbers if a different port than the default port for HTTPS (443) is used.

Replace *local_ip* and *local_mask* with the IP address and subnet mask of a user or subnetwork making requests.

Replace *foreign_ip* and *foreign_mask* with the IP address and subnet mask of a server or subnetwork responding to requests.

The **allow** option causes the security appliance to forward HTTPS traffic without filtering when the primary filtering server is unavailable.

Filtering FTP Requests

You must identify and enable the URL filtering server before enabling FTP filtering.



Note Websense and Smartfilter currently support FTP; older versions of Secure Computing SmartFilter (formerly known as N2H2) did not support FTP filtering.

When the filtering server approves an FTP connection request, the security appliance allows the successful FTP return code to reach originating client. For example, a successful return code is "250: CWD command successful." If the filtering server denies the request, alters the FTP return code to show that the connection was denied. For example, the security appliance changes code 250 to "550 Requested file is prohibited by URL filtering policy."

To enable FTP filtering, enter the following command:

```
hostname(config)# filter ftp port[-port] localIP local_mask foreign_IP foreign_mask
[allow] [interact-block]
```

Replace *port*[*-port*] with a range of port numbers if a different port than the default port for FTP (21) is used.

Replace *local_ip* and *local_mask* with the IP address and subnet mask of a user or subnetwork making requests.

Replace *foreign_ip* and *foreign_mask* with the IP address and subnet mask of a server or subnetwork responding to requests.

The **allow** option causes the security appliance to forward HTTPS traffic without filtering when the primary filtering server is unavailable.

Use the **interact-block** option to prevent interactive FTP sessions that do not provide the entire directory path. An interactive FTP client allows the user to change directories without typing the entire path. For example, the user might enter **cd ./files** instead of **cd /public/files**.

Viewing Filtering Statistics and Configuration

This section describes how to monitor filtering statistics. This section includes the following topics:

- Viewing Filtering Server Statistics, page 22-10
- Viewing Buffer Configuration and Statistics, page 22-11
- Viewing Caching Statistics, page 22-11
- Viewing Filtering Performance Statistics, page 22-11
- Viewing Filtering Configuration, page 22-12

Viewing Filtering Server Statistics

To show information about the filtering server, enter the following command:

```
hostname# show running-config url-server
```

The following is sample output from the **show running-config url-server** command:

```
hostname# show running-config url-server
url-server (outside) vendor n2h2 host 128.107.254.202 port 4005 timeout 5 protocol TCP
```

To show information about the filtering server or to show statistics, enter the following command:

The following is sample output from the **show running-config url-server statistics** command, which shows filtering statistics:

```
hostname# show running-config url-server statistics

Global Statistics:
--------------------
URLs total/allowed/denied         13/3/10
URLs allowed by cache/server      0/3
URLs denied by cache/server       0/10
HTTPSs total/allowed/denied       138/137/1
HTTPSs allowed by cache/server    0/137
HTTPSs denied by cache/server     0/1
FTPs total/allowed/denied         0/0/0
FTPs allowed by cache/server      0/0
FTPs denied by cache/server       0/0
Requests dropped                  0
Server timeouts/retries           0/0
Processed rate average 60s/300s   0/0 requests/second
Denied rate average 60s/300s      0/0 requests/second
Dropped rate average 60s/300s     0/0 requests/second

Server Statistics:
--------------------
10.125.76.20                      UP
  Vendor                          websense
  Port                            15868
  Requests total/allowed/denied   151/140/11
  Server timeouts/retries         0/0
  Responses received              151
  Response time average 60s/300s  0/0

URL Packets Sent and Received Stats:
------------------------------------
Message                 Sent    Received
STATUS_REQUEST          1609    1601
LOOKUP_REQUEST          1526    1526
LOG_REQUEST             0       NA

Errors:
-------
RFC noncompliant GET method     0
URL buffer update failure       0
```

Viewing Buffer Configuration and Statistics

The **show running-config url-block** command displays the number of packets held in the url-block buffer and the number (if any) dropped due to exceeding the buffer limit or retransmission.

The following is sample output from the **show running-config url-block** command:

```
hostname# show running-config url-block
    url-block url-mempool 128
    url-block url-size 4
    url-block block 128
```

This shows the configuration of the URL block buffer.

The following is sample output from the **show url-block block statistics** command:

```
hostname# show running-config url-block block statistics

URL Pending Packet Buffer Stats with max block  128
-----------------------------------------------------
Cumulative number of packets held:              896
Maximum number of packets held (per URL):       3
Current number of packets held (global):        38
Packets dropped due to
       exceeding url-block buffer limit:        7546
       HTTP server retransmission:              10
Number of packets released back to client:      0
```

This shows the URL block statistics.

Viewing Caching Statistics

The following is sample output from the **show url-cache stats** command:

```
hostname# show url-cache stats
URL Filter Cache Stats
----------------------
    Size :     128KB
 Entries :      1724
  In Use :       456
 Lookups :        45
    Hits :         8

This shows how the cache is used.
```

Viewing Filtering Performance Statistics

The following is sample output from the **show perfmon** command:

```
hostname# show perfmon
PERFMON STATS:    Current      Average
Xlates               0/s          0/s
Connections          0/s          2/s
TCP Conns            0/s          2/s
UDP Conns            0/s          0/s
URL Access           0/s          2/s
URL Server Req       0/s          3/s
TCP Fixup            0/s          0/s
TCPIntercept         0/s          0/s
HTTP Fixup           0/s          3/s
```

```
FTP Fixup            0/s          0/s
AAA Authen           0/s          0/s
AAA Author           0/s          0/s
AAA Account          0/s          0/s
```

This shows URL filtering performance statistics, along with other performance statistics. The filtering statistics are shown in the URL Access and URL Server Req rows.

Viewing Filtering Configuration

The following is sample output from the **show running-config filter** command:

```
hostname# show running-config filter
filter url http 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
```



CHAPTER 23

Managing the AIP SSM and CSC SSM

This chapter describes how to configure the adaptive security appliance to support an AIP SSM or a CSC SSM that is installed in the adaptive security appliance.

See Chapter 6, "Configuring Ethernet Settings, Redundant Interfaces, and Subinterfaces" for information about the 4GE SSM for the ASA 5500 series adaptive security appliance.

Note The Cisco PIX 500 series security appliances do not support SSMs. The ASA 5510, ASA 5520, and ASA 5540 series adaptive security appliances support SSMs.

This chapter includes the following sections:

- Managing the AIP SSM, page 23-1
- Managing the CSC SSM, page 23-9
- Checking SSM Status, page 23-18
- Transferring an Image onto an SSM, page 23-19

Managing the AIP SSM

This section includes the following topics:

- AIP SSM Overview, page 23-1
- Sessioning to the AIP SSM, page 23-5
- Configuring the Security Policy on the AIP SSM, page 23-6
- Assigning Virtual Sensors to Security Contexts, page 23-6
- Diverting Traffic to the AIP SSM, page 23-8

AIP SSM Overview

You can install the AIP SSM into an ASA 5500 series adaptive security appliance. The AIP SSM runs advanced IPS software that provides proactive, full-featured intrusion prevention services to stop malicious traffic, including worms and network viruses, before they can affect your network. This section includes the following topics:

- How the AIP SSM Works with the Adaptive Security Appliance, page 23-2

- Operating Modes, page 23-3
- Using Virtual Sensors, page 23-3
- AIP SSM Procedure Overview, page 23-4

How the AIP SSM Works with the Adaptive Security Appliance

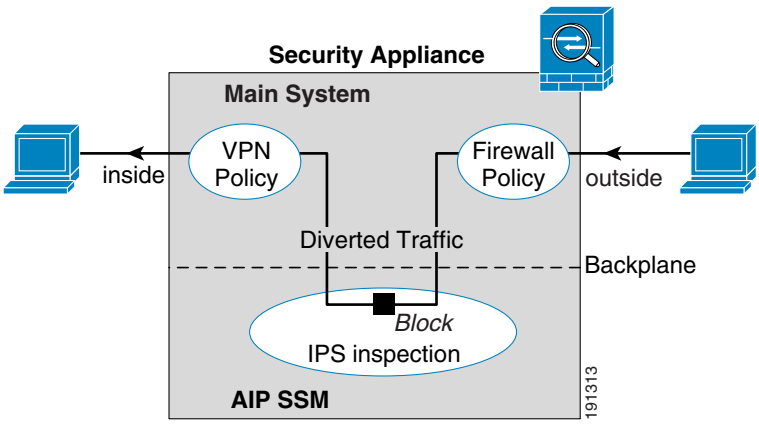
The AIP SSM runs a separate application from the adaptive security appliance. It is, however, integrated into the adaptive security appliance traffic flow. The AIP SSM does not contain any external interfaces itself, other than a management interface. When you identify traffic for IPS inspection on the adaptive security appliance, traffic flows through the adaptive security appliance and the AIP SSM in the following way:

1. Traffic enters the adaptive security appliance.
2. Firewall policies are applied.
3. Traffic is sent to the AIP SSM over the backplane.

   See the "Operating Modes" section on page 23-3 for information about only sending a copy of the traffic to the AIP SSM.
4. The AIP SSM applies its security policy to the traffic, and takes appropriate actions.
5. Valid traffic is sent back to the adaptive security appliance over the backplane; the AIP SSM might block some traffic according to its security policy, and that traffic is not passed on.
6. VPN policies are applied (if configured).
7. Traffic exits the adaptive security appliance.

Figure 23-1 shows the traffic flow when running the AIP SSM in inline mode. In this example, the AIP SSM automatically blocks traffic that it identified as an attack. All other traffic is forwarded through the adaptive security appliance.

Figure 23-1 AIP SSM Traffic Flow in the Adaptive Security Appliance: Inline Mode

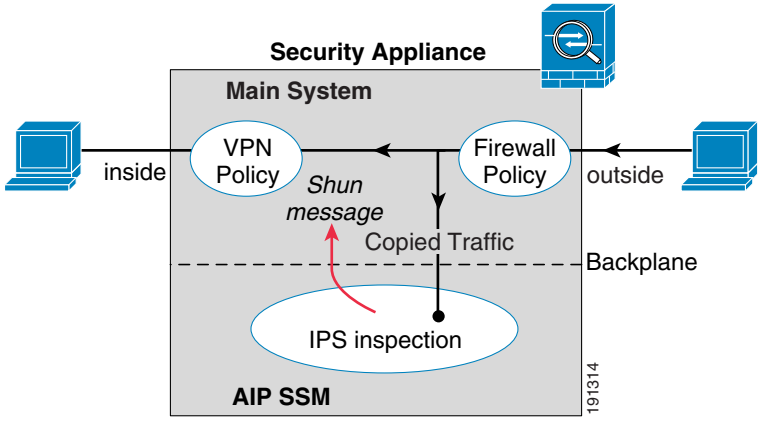


Operating Modes

You can send traffic to the AIP SSM using one of the following modes:

- Inline mode—This mode places the AIP SSM directly in the traffic flow (see Figure 23-1). No traffic that you identified for IPS inspection can continue through the adaptive security appliance without first passing through, and being inspected by, the AIP SSM. This mode is the most secure because every packet that you identify for inspection is analyzed before being allowed through. Also, the AIP SSM can implement a blocking policy on a packet-by-packet basis. This mode, however, can affect throughput.
- Promiscuous mode—This mode sends a duplicate stream of traffic to the AIP SSM. This mode is less secure, but has little impact on traffic throughput. Unlike the inline mode, in promiscuous mode the AIP SSM can only block traffic by instructing the adaptive security appliance to shun the traffic or by resetting a connection on the adaptive security appliance. Also, while the AIP SSM is analyzing the traffic, a small amount of traffic might pass through the adaptive security appliance before the AIP SSM can shun it. Figure 23-2 shows the AIP SSM in promiscuous mode. In this example, the AIP SSM sends a shun message to the security appliance for traffic it identified as a threat.

Figure 23-2 AIP SSM Traffic Flow in the Adaptive Security Appliance: Promiscuous Mode



Using Virtual Sensors

The AIP SSM running IPS software Version 6.0 and above can run multiple virtual sensors, which means you can configure multiple security policies on the AIP SSM. You can assign each context or single mode security appliance to one or more virtual sensors, or you can assign multiple security contexts to the same virtual sensor. See the IPS documentation for more information about virtual sensors, including the maximum number of sensors supported.

Figure 23-3 shows one security context paired with one virtual sensor (in inline mode), while two security contexts share the same virtual sensor.

Figure 23-3 Security Contexts and Virtual Sensors

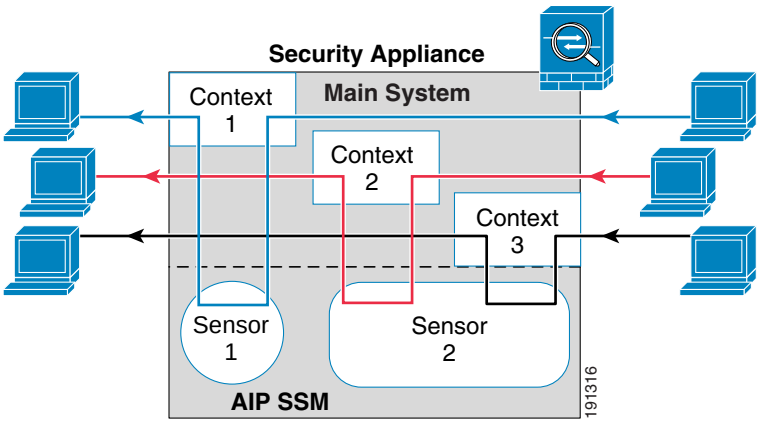
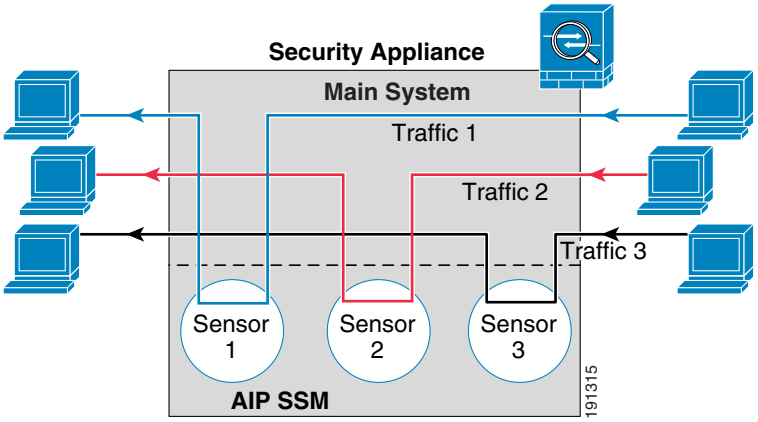


Figure 23-4 shows a single mode security appliance paired with multiple virtual sensors (in inline mode); each defined traffic flow goes to a different sensor.

Figure 23-4 Single Mode Security Appliance with Multiple Virtual Sensors



AIP SSM Procedure Overview

Configuring the AIP SSM is a process that includes configuration of the AIP SSM and then configuration of the ASA 5500 series adaptive security appliance:

1. Session to the AIP SSM from the security appliance. See the "Sessioning to the AIP SSM" section on page 23-5.
2. On the AIP SSM, configure the inspection and protection policy, which determines how to inspect traffic and what to do when an intrusion is detected. Configure the inspection and protection policy for each virtual sensor if you want to run the AIP SSM in multiple sensor mode. See the "Configuring the Security Policy on the AIP SSM" section on page 23-6.
3. On the ASA 5500 series adaptive security appliance in multiple context mode, specify which IPS virtual sensors are available for each context (if you configured virtual sensors). See the "Assigning Virtual Sensors to Security Contexts" section on page 23-6.
4. On the ASA 5500 series adaptive security appliance, identify traffic to divert to the AIP SSM. See the "Diverting Traffic to the AIP SSM" section on page 23-8.

Sessioning to the AIP SSM

To begin configuring the AIP SSM, session to the AIP SSM from the adaptive security appliance. (You can alternatively connect directly to the AIP SSM management interface using SSH or Telnet.)

To session to the AIP SSM from the adaptive security appliance, perform the following steps:

Step 1 To session from the ASA 5500 series adaptive security appliance to the AIP SSM, enter the following command:

```
hostname# session 1

Opening command session with slot 1.
Connected to slot 1. Escape character sequence is 'CTRL-^X'.
```

Step 2 Enter the username and password. The default username and password is “cisco.”

Note The first time you log in to the AIP SSM, you are prompted to change the default password. Passwords must be at least eight characters long and not a word in the dictionary.

```
login: cisco
Password:
Last login: Fri Sep  2 06:21:20 from xxx.xxx.xxx.xxx
***NOTICE***
This product contains cryptographic features and is subject to United States
and local country laws governing import, export, transfer and use. Delivery
of Cisco cryptographic products does not imply third-party authority to import,
export, distribute or use encryption. Importers, exporters, distributors and
users are responsible for compliance with U.S. and local country laws. By using
this product you agree to comply with applicable laws and regulations. If you
are unable to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

***LICENSE NOTICE***
There is no license key installed on the system.
Please go to http://www.cisco.com/go/license
to obtain a new license or install a license.
AIP SSM#
```

Note If you see the preceding license notice (which displays only in some versions of software), you can ignore the message until you need to upgrade the signature files on the AIP SSM. The AIP SSM continues to operate at the current signature level until a valid license key is installed. You can install the license key at a later time. The license key does not affect the current functionality of the AIP SSM.

Configuring the Security Policy on the AIP SSM

On the AIP SSM, to configure the inspection and protection policy, which determines how to inspect traffic and what to do when an intrusion is detected, perform the following steps. To session from the security appliance to the AIP SSM, see the "Sessioning to the AIP SSM" section on page 23-5.

Step 1 To run the setup utility for initial configuration of the AIP SSM, enter the following command:

```
sensor# setup
```

Step 2 Configure the IPS security policy. If you configure virtual sensors in IPS Version 6.0 or above, you identify one of the sensors as the default. If the ASA 5500 series adaptive security appliance does not specify a virtual sensor name in its configuration, the default sensor is used.

Because the IPS software that runs on the AIP SSM is beyond the scope of this document, detailed configuration information is available at the following URL:

http://www.cisco.com/en/US/products/hw/vpndevc/ps4077/tsd_products_support_series_home.html

Step 3 When you are done configuring the AIP SSM, exit the IPS software by entering the following command:

```
sensor# exit
```

If you sessioned to the AIP SSM from the security appliance, you return to the security appliance prompt.

Assigning Virtual Sensors to Security Contexts

If the security appliance is in multiple context mode, then you can assign one or more IPS virtual sensors to each context. Then, when you configure the context to send traffic to the AIP SSM, you can specify a sensor that is assigned to the context; you cannot specify a sensor that you did not assign to the context. If you do not assign any sensors to a context, then the default sensor configured on the AIP SSM is used. You can assign the same sensor to multiple contexts.

Note You do not need to be in multiple context mode to use virtual sensors; you can be in single mode and use different sensors for different traffic flows.

To assign one or more sensors to a security context, perform the following steps:

Step 1 To enter context configuration mode, enter the following command in the system execution space:

```
hostname(config)# context name
hostname(config-ctx)#
```

For more information about configuring contexts, see the "Configuring a Security Context" section on page 7-7.

Step 2 To assign a virtual sensor to the context, enter the following command:

```
hostname(config-ctx)# allocate-ips sensor_name [mapped_name] [default]
```

Enter this command for each sensor you want to assign to the context.

The *sensor _name* argument is the sensor name configured on the AIP SSM. To view the sensors that are configured on the AIP SSM, enter **allocate-ips ?**. All available sensors are listed. You can also enter the **show ips** command. In the system execution space, the **show ips** command lists all available sensors; if you enter it in the context, it shows the sensors you already assigned to the context. If you specify a sensor name that does not yet exist on the AIP SSM, you get an error, but the **allocate-ips** command is entered as is. Until you create a sensor of that name on the AIP SSM, the context assumes the sensor is down.

Use the *mapped_name* argument as an alias for the sensor name that can be used within the context instead of the actual sensor name. If you do not specify a mapped name, the sensor name is used within the context. For security purposes, you might not want the context administrator to know which sensors are being used by the context. Or you might want to genericize the context configuration. For example, if you want all contexts to use sensors called "sensor1" and "sensor2," then you can map the "highsec" and "lowsec" senors to sensor1 and sensor2 in context A, but map the "medsec" and "lowsec" sensors to sensor1 and sensor2 in context B.

The **default** keyword sets one sensor per context as the default sensor; if the context configuration does not specify a sensor name, the context uses this default sensor. You can only configure one default sensor per context. If you want to change the default sensor, enter the **no allocate-ips** *sensor_name* command to remove the current default sensor before you allocate a new default sensor. If you do not specify a sensor as the default, and the context configuration does not include a sensor name, then traffic uses the default sensor on the AIP SSM.

Step 3 Repeat Step 1 and Step 2 for each context.

Step 4 To configure the context IPS policy, change to the context execution space using the following command:

```
hostname(config-ctx)# changeto context context_name
```

where the *context_name* argument is the name of the context you want to configure. Change to each context to configure the IPS security policy as described in "Diverting Traffic to the AIP SSM" section on page 23-8.

The following example assigns sensor1 and sensor2 to context A, and sensor1 and sensor3 to context B. Both contexts map the sensor names to "ips1" and "ips2." In context A, sensor1 is set as the default sensor, but in context B, no default is set so the default that is configured on the AIP SSM is used.

```
hostname(config-ctx)# context A
hostname(config-ctx)# allocate-interface gigabitethernet0/0.100 int1
hostname(config-ctx)# allocate-interface gigabitethernet0/0.102 int2
hostname(config-ctx)# allocate-interface gigabitethernet0/0.110-gigabitethernet0/0.115
int3-int8
hostname(config-ctx)# allocate-ips sensor1 ips1 default
hostname(config-ctx)# allocate-ips sensor2 ips2
hostname(config-ctx)# config-url ftp://user1:passw0rd@10.1.1.1/configlets/test.cfg
hostname(config-ctx)# member gold

hostname(config-ctx)# context sample
hostname(config-ctx)# allocate-interface gigabitethernet0/1.200 int1
hostname(config-ctx)# allocate-interface gigabitethernet0/1.212 int2
hostname(config-ctx)# allocate-interface gigabitethernet0/1.230-gigabitethernet0/1.235
int3-int8
hostname(config-ctx)# allocate-ips sensor1 ips1
hostname(config-ctx)# allocate-ips sensor3 ips2
hostname(config-ctx)# config-url ftp://user1:passw0rd@10.1.1.1/configlets/sample.cfg
hostname(config-ctx)# member silver

hostname(config-ctx)# changeto context A
...
```

Diverting Traffic to the AIP SSM

To identify traffic to divert from the adaptive security appliance to the AIP SSM, perform the following steps. In multiple context mode, perform these steps in each context execution space.

Step 1 To identify the traffic that you want to be inspected by the AIP SSM, add one or more class maps using the **class-map** command according to the "Creating a Layer 3/4 Class Map for Through Traffic" section on page 16-5.

For example, you can match all traffic using the following commands:

```
hostname(config)# class-map IPS
hostname(config-cmap)# match any
```

To match specific traffic, you can match an access list:

```
hostname(config)# access list IPS extended permit ip any 10.1.1.1 255.255.255.255
hostname(config)# class-map IPS
hostname(config-cmap)# match access-list IPS
```

Step 2 To add or edit a policy map that sets the action to divert traffic to the AIP SSM, enter the following commands:

```
hostname(config)# policy-map name
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

where the *class_map_name* is the class map from Step 1.

For example:

```
hostname(config)# policy-map IPS
hostname(config-pmap)# class IPS
```

Step 3 To divert the traffic to the AIP SSM, enter the following command:

```
hostname(config-pmap-c)# ips {inline | promiscuous} {fail-close | fail-open} [sensor
{sensor_name | mapped_name}]
```

where the **inline** and **promiscuous** keywords control the operating mode of the AIP SSM. See the "Operating Modes" section on page 23-3 for more details.

The **fail-close** keyword sets the adaptive security appliance to block all traffic if the AIP SSM is unavailable.

The **fail-open** keyword sets the adaptive security appliance to allow all traffic through, uninspected, if the AIP SSM is unavailable.

If you use virtual sensors on the AIP SSM, you can specify a sensor name using the **sensor** *sensor_name* argument. To see available sensor names, enter the **ips ... sensor ?** command. Available sensors are listed. You can also use the **show ips** command. If you use multiple context mode on the security appliance, you can only specify sensors that you assigned to the context (see the "Assigning Virtual Sensors to Security Contexts" section on page 23-6). Use the *mapped_name* if configured in the context. If you do not specify a sensor name, then the traffic uses the default sensor. In multiple context mode, you can specify a default sensor for the context. In single mode or if you do not specify a default sensor in multiple mode, the traffic uses the default sensor that is set on the AIP SSM. If you enter a name that does not yet exist on the AIP SSM, you get an error, and the command is rejected.

Step 4 (Optional) To divert another class of traffic to the AIP SSM, and set the IPS policy, enter the following commands:

```
hostname(config-pmap-c)# class class_map_name2
hostname(config-pmap-c)# ips {inline | promiscuous} {fail-close | fail-open} [sensor
sensor_name]
```

where the *class_map_name2* argument is the name of a separate class map on which you want to perform IPS inspection. See Step 3 for information about the command options. See the "Information About Layer 3/4 Policy Maps" section on page 16-17 for detailed information about how the order of classes matters within a policy map. Traffic cannot match more than one class map for the same action type; so if you want network A to go to sensorA, but want all other traffic to go to sensorB, then you need to enter the **class** command for network A before you enter the **class** command for all traffic; otherwise all traffic (including network A) will match the first **class** command, and will be sent to sensorB.

Step 5 To activate the policy map on one or more interfaces, enter the following command:

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface interface_ID]
hostname
```

where *policy_map_name* is the policy map you configured in Step 2. To apply the policy map to traffic on all the interfaces, use the **global** keyword. To apply the policy map to traffic on a specific interface, use the **interface** *interface_ID* option, where *interface_ID* is the name assigned to the interface with the **nameif** command.

Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.

The following example diverts all IP traffic to the AIP SSM in promiscuous mode, and blocks all IP traffic if the AIP SSM card fails for any reason:

```
hostname(config)# access-list IPS permit ip any any
hostname(config)# class-map my-ips-class
hostname(config-cmap)# match access-list IPS
hostname(config-cmap)# policy-map my-ips-policy
hostname(config-pmap)# class my-ips-class
hostname(config-pmap-c)# ips promiscuous fail-close
hostname(config-pmap-c)# service-policy my-ips-policy global
```

The following example diverts all IP traffic destined for the 10.1.1.0 network and the 10.2.1.0 network to the AIP SSM in inline mode, and allows all traffic through if the AIP SSM card fails for any reason. For the my-ips-class traffic, sensor1 is used; for the my-ips-class2 traffic, sensor2 is used.

```
hostname(config)# access-list my-ips-acl permit ip any 10.1.1.0 255.255.255.0
hostname(config)# access-list my-ips-acl2 permit ip any 10.2.1.0 255.255.255.0
hostname(config)# class-map my-ips-class
hostname(config-cmap)# match access-list my-ips-acl
hostname(config)# class-map my-ips-class2
hostname(config-cmap)# match access-list my-ips-acl2
hostname(config-cmap)# policy-map my-ips-policy
hostname(config-pmap)# class my-ips-class
hostname(config-pmap-c)# ips inline fail-open sensor sensor1
hostname(config-pmap)# class my-ips-class2
hostname(config-pmap-c)# ips inline fail-open sensor sensor2
hostname(config-pmap-c)# service-policy my-ips-policy interface outside
```

Managing the CSC SSM

This section includes the following topics:

- About the CSC SSM, page 23-10
- Getting Started with the CSC SSM, page 23-12
- Determining What Traffic to Scan, page 23-13
- Limiting Connections Through the CSC SSM, page 23-15
- Diverting Traffic to the CSC SSM, page 23-16

About the CSC SSM

The ASA 5500 series adaptive security appliance supports the CSC SSM, which runs Content Security and Control software. The CSC SSM provides protection against viruses, spyware, spam, and other unwanted traffic by scanning the FTP, HTTP, POP3, and SMTP packets that you configure the adaptive security appliance to send to it.

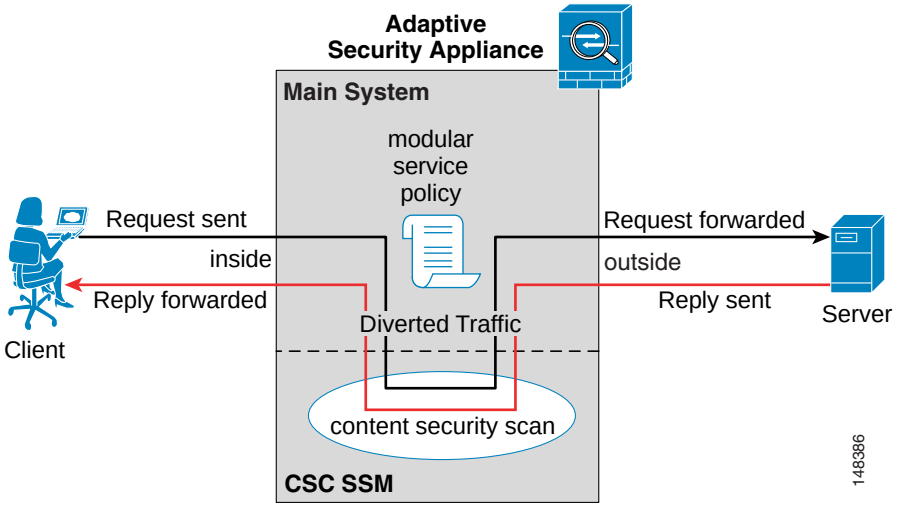
Figure 23-5 illustrates the flow of traffic through an adaptive security appliance that has the following:

- A CSC SSM installed and configured.
- A service policy that determines what traffic is diverted to the CSC SSM for scanning.

In this example, the client could be a network user who is accessing a website, downloading files from an FTP server, or retrieving mail from a POP3 server. SMTP scans differ in that you should configure the adaptive security appliance to scan traffic sent from the outside to SMTP servers protected by the adaptive security appliance.

Note The CSC SSM can scan FTP file transfers only when FTP inspection is enabled on the adaptive security appliance. By default, FTP inspection is enabled.

Figure 23-5 Flow of Scanned Traffic with CSC SSM



You use ASDM for system setup and monitoring of the CSC SSM. For advanced configuration of content security policies in the CSC SSM software, you access the web-based GUI for the CSC SSM by clicking links within ASDM. For instructions on use of the CSC SSM GUI, see the *Trend Micro InterScan for Cisco CSC SSM Administrator Guide*.

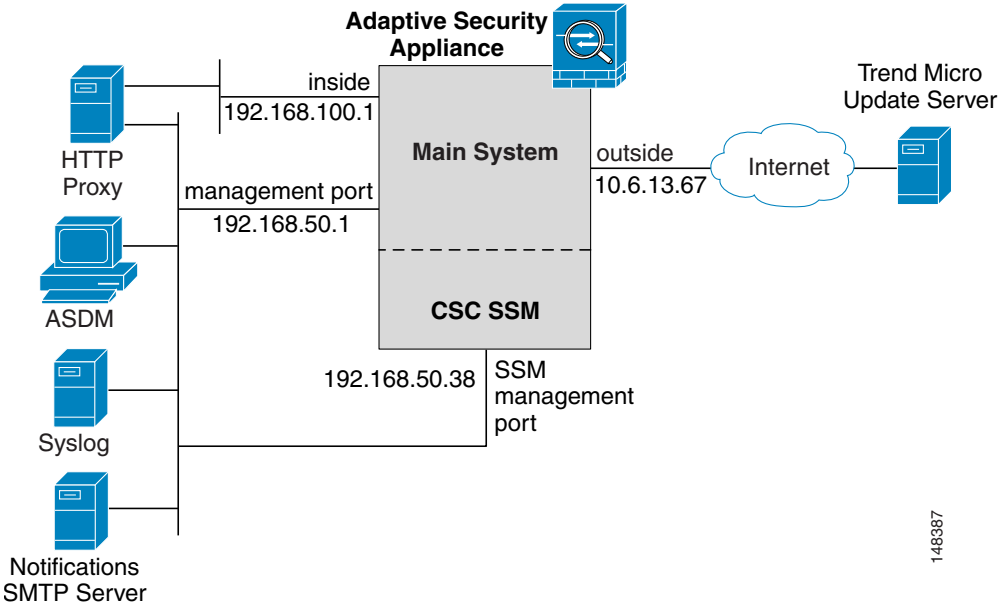
Note ASDM and the CSC SSM maintain separate passwords. You can configure their passwords to be identical; however, changing one of these two passwords does not affect the other password.

The connection between the host running ASDM and the adaptive security appliance is made through a management port on the adaptive security appliance. The connection to the CSC SSM GUI is made through the SSM management port. Because these two connections are required to manage the CSC SSM, any host running ASDM must be able to reach the IP address of both the adaptive security appliance management port and the SSM management port.

Figure 23-6 shows an adaptive security appliance with a CSC SSM that is connected to a dedicated management network. While use of a dedicated management network is not required, we recommend it. Of particular interest are the following:

- An HTTP proxy server is connected to the inside network and to the management network. This HTTP proxy server enables the CSC SSM to contact the Trend Micro update server.
- The management port of the adaptive security appliance is connected to the management network. To permit management of the adaptive security appliance and the CSC SSM, hosts running ASDM must be connected to the management network.
- The management network includes an SMTP server for e-mail notifications for the CSC SSM and a syslog server to which the CSC SSM can send system log messages.

Figure 23-6 CSC SSM Deployment with a Management Network



The CSC SSM cannot support Stateful Failover because the CSC SSM does not maintain connection information, and therefore cannot provide the failover unit with the required information for Stateful Failover. The connections that a CSC SSM is scanning are dropped when the security appliance in which the CSC SSM is installed fails. When the standby adaptive security appliance becomes active, it will forward the scanned traffic to the CSC SSM and the connections will be reset.

Getting Started with the CSC SSM

Before you receive the security benefits provided by a CSC SSM, you must perform several steps beyond hardware installation of the SSM. This procedure provides an overview of those steps.

To configure the adaptive security appliance and the CSC SSM, follow these steps:

Step 1 If the CSC SSM did not come pre-installed in a Cisco ASA 5500 series adaptive security appliance, install it and connect a network cable to the management port of the SSM. For assistance with installation and connecting the SSM, see the *Cisco ASA 5500 Series Hardware Installation Guide*.

The management port of the CSC SSM must be connected to your network to allow management of and automatic updates to the CSC SSM software. Additionally, the CSC SSM uses the management port for e-mail notifications and system log messaging.

Step 2 With the CSC SSM, you should have received a Product Authorization Key (PAK). Use the PAK to register the CSC SSM at the following URL.

`http://www.cisco.com/go/license`

After you register, you will receive activation keys by e-mail. The activation keys are required before you can complete Step 6

Step 3 Gather the following information for use in Step 6.

- Activation keys, received after completing Step 2.
- The CSC SSM management port IP address, netmask, and gateway IP address.

Note The CSC SSM management port IP address must be accessible by the hosts used to run ASDM. The IP addresses for the CSC SSM management port and the adaptive security appliance management interface can be in different subnets.

- DNS server IP address.
- HTTP proxy server IP address (needed only if your security policies require the use of a proxy server for HTTP access to the Internet).
- Domain name and hostname for the CSC SSM.
- An e-mail address and an SMTP server IP address and port number for e-mail notifications.
- IP addresses of hosts or networks allowed to manage the CSC SSM.
- Password for the CSC SSM.

Step 4 In a web browser, access ASDM for the adaptive security appliance in which the CSC SSM is installed.

Note If you are accessing ASDM for the first time, see the *Cisco ASA 5500 Series Adaptive Security Appliance Getting Started Guide* for assistance with the Startup Wizard.

For more information about enabling ASDM access, see the "Allowing HTTPS Access for ASDM" section on page 42-3.

Step 5 Verify time settings on the adaptive security appliance. Time setting accuracy is important for logging of security events and for automatic updates of CSC SSM software.

- If you manually control time settings, verify the clock settings, including time zone. Choose **Configuration > Properties > Device Administration > Clock**.

- If you are using NTP, verify the NTP configuration. Choose **Configuration > Properties > Device Administration > NTP**.

Step 6 To access the ASDM GUI in a supported web browser and on the Home page, click the **Content Security** tab. In ASDM, run the CSC Setup Wizard. To access the CSC Setup Wizard, choose **Configuration > Trend Micro Content Security > CSC Setup > Wizard Setup > Launch Setup Wizard**. The CSC Setup Wizard appears. For assistance with the CSC Setup Wizard, click the **Help** button.

Note If you are accessing ASDM for the first time, see the *Cisco ASA 5500 Series Adaptive Security Appliance Getting Started Guide* for assistance with the Startup Wizard.

Step 7 On the ASA 5500 series adaptive security appliance, identify traffic to divert to the CSC SSM (see the "Diverting Traffic to the CSC SSM" section on page 23-16).

Step 8 (Optional) Review the default content security policies in the CSC SSM GUI. The default content security policies are suitable for most implementations. Before you modify them or enter advanced configuration settings, review the *Trend Micro InterScan for Cisco CSC SSM Administrator Guide*.

You review the content security policies by viewing the enabled features in the CSC SSM GUI. The availability of features depends on the license level you have purchased. By default, all features included in the license you have purchased are enabled.

With a Base License, the features enabled by default are SMTP virus scanning, POP3 virus scanning and content filtering, webmail virus scanning, HTTP file blocking, FTP virus scanning and file blocking, logging, and automatic updates.

With a Plus License, the additional features enabled by default are SMTP anti-spam, SMTP content filtering, POP3 anti-spam, URL blocking, and URL filtering.

To access the CSC SSM GUI, in ASDM choose **Configuration > Trend Micro Content Security**, and then select one of the following: **Web**, **Mail**, **File Transfer**, or **Updates**. The links on these panes, beginning with the word "Configure," open the CSC SSM GUI.

Determining What Traffic to Scan

The CSC SSM can scan FTP, HTTP, POP3, and SMTP traffic only when the destination port of the packet requesting the connection is the well-known port for the specified protocol. The CSC SSM can scan only the following connections:

- FTP connections opened to TCP port 21.
- HTTP connections opened to TCP port 80.
- POP3 connections opened to TCP port 110.
- SMTP connections opened to TCP port 25.

You can choose to scan traffic for all of these protocols or any combination of them. For example, if you do not allow network users to receive POP3 e-mail, do not configure the adaptive security appliance to divert POP3 traffic to the CSC SSM. Instead, block this traffic.

To maximize performance of the adaptive security appliance and the CSC SSM, divert to the CSC SSM only the traffic that you want the CSC SSM to scan. Needlessly diverting traffic that you do not want to scan, such as traffic between a trusted source and destination, can adversely affect network performance.

To enable traffic scanning with the CSC SSM, use the **csc** command, which must be part of a service policy. Service policies can be applied globally or to specific interfaces; therefore, you can enable the **csc** command globally or for specific interfaces.

Adding the **csc** command to your global policy ensures that all unencrypted connections through the adaptive security appliance are scanned by the CSC SSM; however, this setting may mean that traffic from trusted sources is needlessly scanned.

If you enable the **csc** command in interface-specific service policies, it is bi-directional. Bi-directionality means that when the adaptive security appliance opens a new connection, if the **csc** command is active on either the inbound or the outbound interface of the connection and the class map for the policy identifies traffic for scanning, the adaptive security appliance diverts this traffic to the CSC SSM.

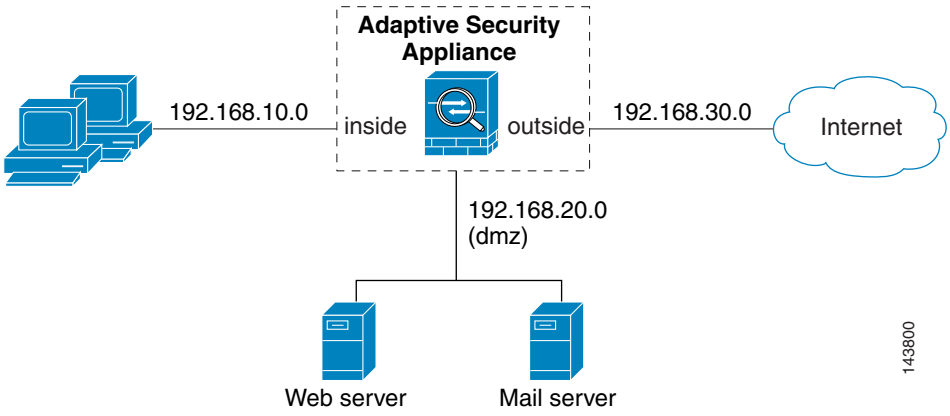
However, bi-directionality also means that if you divert any of the supported traffic types that cross a given interface to the CSC SSM, it is probably performing unnecessary scans on traffic from your trusted inside networks. For example, URLs and files requested from web servers on a DMZ network are unlikely to pose content security risks to hosts on an inside network, and you probably do not want the adaptive security appliance to divert this traffic to the CSC SSM.

Therefore, we recommend using access lists to further limit the traffic selected by the class maps of CSC SSM service policies. Specifically, use access lists that match the following:

- HTTP connections to outside networks.
- FTP connections from clients inside the adaptive security appliance to servers outside the adaptive security appliance.
- POP3 connections from clients inside the security appliance to servers outside the adaptive security appliance.
- Incoming SMTP connections destined to inside mail servers.

In Figure 23-7, the adaptive security appliance should be configured to divert traffic to CSC SSM requests from clients on the inside network for HTTP, FTP, and POP3 connections to the outside network and incoming SMTP connections from outside hosts to the mail server on the DMZ network. HTTP requests from the inside network to the web server on the DMZ network should not be scanned.

Figure 23-7 Common Network Configuration for CSC SSM Scanning



To identify the traffic that you want to scan, you can configure the adaptive security appliance in different ways. One approach is to define two service policies, one on the inside interface and the other on the outside interface, each with an access list that matches traffic to be scanned. The following access list can be used on the policy applied to the inside interface:

```
access-list csc_out permit tcp 192.168.10.0 255.255.255.0 any eq 21
access-list csc_out deny tcp 192.168.10.0 255.255.255.0 192.168.20.0 255.255.255.0 eq 80
access-list csc_out permit tcp 192.168.10.0 255.255.255.0 any eq 80
access-list csc_out permit tcp 192.168.10.0 255.255.255.0 any eq 110
```

As previously mentioned, policies applying the **csc** command to a specific interface are effective on both ingress and egress traffic. However, by specifying 192.168.10.0 as the source network in the csc_out access list, the policy applied to the inside interface matches only connections initiated by the hosts on the inside network. Notice also that the second ACE of the access list contains the **deny** keyword. This ACE does not mean the adaptive security appliance blocks traffic sent from the 192.168.10.0 network to TCP port 80 on the 192.168.20.0 network. Instead, the ACE exempts the traffic from being matched by the policy map and thus prevents the adaptive security appliance from sending the traffic to the CSC SSM.

You can use **deny** keywords in an access list to exempt connections with trusted external hosts from being scanned. For example, to reduce the load on the CSC SSM, you might want to exempt HTTP traffic to a well-known, trusted site. If the web server at this site has the IP address 209.165.201.7, you could add the following ACE to the csc_out access list to exclude HTTP connections between the trusted external web server and inside hosts from being scanned by the CSC SSM:

```
access-list csc_out deny tcp 192.168.10.0 255.255.255.0 209.165.201.7 255.255.255.255 eq 80
```

The second policy in this example, applied to the outside interface, could use the following access list:

```
access-list csc_in permit tcp any 192.168.20.0 255.255.255.0 eq 25
```

This access list matches inbound SMTP connections from any external host to any host on the DMZ network. The policy applied to the outside interface would therefore ensure that incoming SMTP e-mail would be diverted to the CSC SSM for scanning. However, the policy would not match SMTP connections from hosts on the inside network to the mail server on the DMZ network, because those connections never use the outside interface.

If the web server on the DMZ network receives files uploaded by HTTP from external hosts, you could add the following ACE to the csc_in access list to use the CSC SSM to protect the web server from infected files:

```
access-list csc_in permit tcp any 192.168.20.0 255.255.255.0 eq 80
```

For a service policy configuration using the access lists in this section, see Example 23-1.

Limiting Connections Through the CSC SSM

The adaptive security appliance can prevent the CSC SSM and the destinations of connections it scans from accepting or even receiving requests for more connections than desired. It can do so for embryonic connections or fully established connections. Also, you can specify limits for all clients included in a class-map and per-client limits. The **set connection** command lets you configure limits for embryonic connections or fully established connections.

Also, you can specify limits for all clients included in a class-map and per-client limits. The **per-client-embryonic-max** and **per-client-max** parameters limit the maximum number of connections that individual clients can open. If a client uses more network resources simultaneously than is desired, you can use these parameters to limit the number of connections that the adaptive security appliance allows for each client.

DoS attacks seek to disrupt networks by overwhelming the capacity of key hosts with connections or requests for connections. You can use the **set connection** command to thwart DoS attacks. After you configure a per-client maximum that can be supported by hosts likely to be attacked, malicious clients will be unable to overwhelm hosts on protected networks.

For use of the **set connection** command to protect the CSC SSM and the destinations of connections it scans, see the "Diverting Traffic to the CSC SSM" section on page 23-16.

Diverting Traffic to the CSC SSM

You use Modular Policy Framework commands to configure the adaptive security appliance to divert traffic to the CSC SSM. Before configuring the adaptive security appliance to divert traffic to the CSC SSM, review Chapter 16, "Using Modular Policy Framework," which introduces Modular Policy Framework concepts and common commands.

To identify traffic to divert from the adaptive security appliance to the CSC SSM, perform the following steps:

Step 1 Create an access list that matches the traffic you want scanned by the CSC SSM with the **access-list extended** command. Create as many ACEs as are needed to match all the traffic. For example, to specify FTP, HTTP, POP3, and SMTP traffic, you need four ACEs. For guidance on identifying the traffic you want to scan, see the "Determining What Traffic to Scan" section on page 23-13.

Step 2 Create a class map to identify the traffic that should be diverted to the CSC SSM with the **class-map** command:

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

where *class_map_name* is the name of the traffic class. When you enter the **class-map** command, the CLI enters class map configuration mode.

Step 3 With the access list you created in Step 1, use a **match access-list** command to identify the traffic to be scanned:

```
hostname(config-cmap)# match access-list acl-name
```

where *acl-name* is the name of the access list.

Step 4 Create a policy map or modify an existing policy map that you want to use to send traffic to the CSC SSM with the **policy-map** command:

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

where *policy_map_name* is the name of the policy map. The CLI enters the policy map configuration mode and the prompt changes accordingly.

Step 5 Specify the class map, created in Step 2, that identifies the traffic to be scanned. Use the **class** command to do so, as follows.

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

where *class_map_name* is the name of the class map you created in Step 2. The CLI enters the policy map class configuration mode and the prompt changes accordingly.

Step 6 If you want to enforce a per-client limit for simultaneous connections that the adaptive security appliance diverts to the CSC SSM, use the **set connection** command, as follows:

```
hostname(config-pmap-c)# set connection per-client-max n
```

where *n* is the maximum simultaneous connections the adaptive security appliance will allow per client. This command prevents a single client from abusing the services of the CSC SSM or any server protected by the SSM, including prevention of attempts at DoS attacks on HTTP, FTP, POP3, or SMTP servers that the CSC SSM protects.

Step 7 Assign the traffic identified by the class map as traffic to be sent to the CSC SSM with the **csc** command:

```
hostname(config-pmap-c)# csc {fail-close | fail-open}
```

The **fail-close** and **fail-open** keywords control how the adaptive security appliance handles traffic when the CSC SSM is unavailable. For more information about the operating modes and failure behavior, see the "About the CSC SSM" section on page 23-10.

Step 8 Apply the policy map globally or to a specific interface with the **service-policy** command:

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface interface_ID]
```

where *policy_map_name* is the policy map you configured in Step 4. To apply the policy map to traffic on all the interfaces, use the **global** keyword.To apply the policy map to traffic on a specific interface, use the **interface** *interface_ID* option, where *interface_ID* is the name assigned to the interface with the **nameif** command.

Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.

The adaptive security appliance begins diverting traffic to the CSC SSM as specified.

Example 23-1 is based on the network shown in Figure 23-7 and shows the creation of two service policies:

- The first policy, csc_out_policy, is applied to the inside interface and uses the csc_out access list to ensure that all outbound requests for FTP and POP3 are scanned. The csc_out access list also ensures that HTTP connections from inside to networks on the outside interface are scanned, but it includes a deny ACE to exclude HTTP connections from inside to servers on the DMZ network.
- The second policy, csc_in_policy, is applied to the outside interface and uses the csc_in access list to ensure that requests for SMTP and HTTP originating on the outside interface and destined for the DMZ network are scanned by the CSC SSM. Scanning HTTP requests protects the web server from HTTP file uploads.

Example 23-1 Service Policies for a Common CSC SSM Scanning Scenario

```
hostname(config)# access-list csc_out permit tcp 192.168.10.0 255.255.255.0 any eq 21
hostname(config)# access-list csc_out deny tcp 192.168.10.0 255.255.255.0 192.168.20.0 255.255.255.0 eq 80
hostname(config)# access-list csc_out permit tcp 192.168.10.0 255.255.255.0 any eq 80
hostname(config)# access-list csc_out permit tcp 192.168.10.0 255.255.255.0 any eq 110

hostname(config)# class-map csc_outbound_class
hostname(config-cmap)# match access-list csc_out

hostname(config-cmap)# policy-map csc_out_policy
hostname(config-pmap)# class csc_outbound_class
hostname(config-pmap-c)# csc fail-close

hostname(config-pmap-c)# service-policy csc_out_policy interface inside

hostname(config)# access-list csc_in permit tcp any 192.168.20.0 255.255.255.0 eq 25
hostname(config)# access-list csc_in permit tcp any 192.168.20.0 255.255.255.0 eq 80

hostname(config)# class-map csc_inbound_class
```

```
hostname(config-cmap)# match access-list csc_in

hostname(config-cmap)# policy-map csc_in_policy
hostname(config-pmap)# class csc_inbound_class
hostname(config-pmap-c)# csc fail-close

hostname(config-pmap-c)# service-policy csc_in_policy interface outside
```

Note FTP inspection must be enabled for the CSC SSM to scan files transferred by FTP. FTP inspection is enabled by default.

Checking SSM Status

To check the status of an SSM, use the **show module** command.

The following is sample output from the **show module** command on an adaptive security appliance with a CSC SSM installed. The Status field indicates the operational status of the SSM. An SSM operating normally has a status of "Up" in the output of the **show module** command. While the adaptive security appliance transfers an application image to the SSM, the Status field in the output reads "Recover." For more information about possible statuses, see the entry for the **show module** command in the *Cisco Security Appliance Command Reference.*

```
hostname# show module 1
Mod Card Type                                    Model              Serial No.
--- -------------------------------------------- ------------------ -----------
 0 ASA 5520 Adaptive Security Appliance         ASA5520            P3000000034
 1 ASA 5500 Series Security Services Module-20  ASA-SSM-20         0

Mod MAC Address Range                 Hw Version   Fw Version   Sw Version
--- --------------------------------- ------------ ------------ ---------------
0 000b.fcf8.c30d to 000b.fcf8.c311  1.0      1.0(10)0     7.1(0)1
1 000b.fcf8.012c to 000b.fcf8.012c  1.0      1.0(10)0     Trend Micro InterScan
                                                          Security Module Version 5.0

Mod SSM Application Name           SSM Application Version
--- ------------------------------ --------------------------
 1 Trend Micro InterScan Security Version 5.0

Mod Status             Data Plane Status     Compatibility
--- ------------------ --------------------- -------------
 0 Up Sys              Not Applicable
 1 Up                  Up
```

The argument **1**, at the end of the command, is the slot number occupied by the SSM. If you do not know the slot number, you can omit it and see information about all modules, including the adaptive security appliance, which is considered to occupy slot 0 (zero).

Use the **details** keyword to view additional information for the SSM.

The following is sample output from the **show module details** command on an adaptive security appliance with a CSC SSM installed.

```
hostname# show module 1 details
Getting details from the Service Module, please wait...
ASA 5500 Series Security Services Module-20
Model:              ASA-SSM-20
Hardware version:   1.0
Serial Number:      0
```

```
Firmware version:   1.0(10)0
Software version:   Trend Micro InterScan Security Module Version 5.0
App. name:          Trend Micro InterScan Security Module
App. version:       Version 5.0
Data plane Status:  Up
Status:             Up
HTTP Service:       Up
Mail Service:       Up
FTP Service:       Up
Activated:          Yes
Mgmt IP addr:       10.23.62.92
Mgmt web port:      8443
```

Transferring an Image onto an SSM

For an intelligent SSM, such as the AIP SSM or CSC SSM, you can transfer application images from a TFTP server to the SSM. This process supports upgrade images and maintenance images.

Note If you are upgrading the application on the SSM, the SSM application may support backup of its configuration. If you do not back up the configuration of the SSM application, it is lost when you transfer an image onto the SSM. For more information about how the SSM supports backups, see the documentation for the specified SSM.

To transfer an image onto an intelligent SSM, perform the following steps:

Step 1 Create or modify a recovery configuration for the SSM.

a. Determine if there is a recovery configuration for the SSM. Use the **show module** command with the **recover** keyword:

```
hostname# show module slot recover
```

where *slot* is the slot number occupied by the SSM.

If the **recover** keyword is not valid, a recovery configuration does not exist. This keyword is available only when a recovery configuration exists for the SSM.

Note When the adaptive security appliance operates in multiple context mode, the **configure** keyword is available only in the system context.

If a recovery configuration exists for the SSM, the adaptive security appliance displays it. Examine the recovery configuration closely to ensure that it is correct, particularly the Image URL field. The following is sample output from the **show module recover** command for an SSM in slot 1.

```
hostname# show module 1 recover
Module 1 recover parameters. . .
Boot Recovery Image: Yes
Image URL:           tftp://10.21.18.1/ids-oldimg
Port IP Address:     10.1.2.10
Port Mask:          255.255.255.0
Gateway IP Address:  10.1.2.254
```

b. To create or modify the recovery configuration, use the **hw-module module recover** command with the **configure** keyword:

```
hostname# hw-module module slot recover configure
```

where *slot* is the slot number occupied by the SSM.

- Complete the prompts as applicable. If you are modifying a configuration, you can keep the previously configured value by pressing **Enter**. The following example shows the prompts. For more information about them, see the entry for the **hw-module module recover** command in the *Cisco Security Appliance Command Reference.*

```
Image URL [tftp://0.0.0.0/]:
Port IP Address [0.0.0.0]:
VLAN ID [0]:
Gateway IP Address [0.0.0.0]:
```

Note Be sure the TFTP server you specify can transfer files up to 60 MB in size. Also, be sure the TFTP server can connect to the management port IP address that you specify for the SSM.

After you complete the series of prompts, the adaptive security appliance is ready to transfer the image that it finds to the SSM at the specified URL.

Step 2 To transfer the image from the TFTP server to the SSM and restart the SSM, use the **hw-module module recover** command with the **boot** keyword:

```
hostname# hw-module module slot recover boot
```

where *slot* is the slot number occupied by the SSM.

Step 3 Check the progress of the image transfer and SSM restart process with the **show module** command. For details, see the "Checking SSM Status" section on page 23-18.

When the adaptive security appliance completes the image transfer and restarts the SSM, the newly transferred image is running.

Note If the SSM supports configuration backups and you want to restore the configuration of the application running on the SSM, see the documentation of the specified SSM for details.



CHAPTER 24

Preventing Network Attacks

This chapter describes how to prevent network attacks by configuring threat detection, TCP normalization, limiting of TCP and UDP connections, and many other protection features.

This chapter includes the following sections:

- Configuring Threat Detection, page 24-1
- Configuring TCP Normalization, page 24-12
- Configuring Connection Limits and Timeouts, page 24-17
- Preventing IP Spoofing, page 24-21
- Configuring the Fragment Size, page 24-22
- Blocking Unwanted Connections, page 24-22
- Configuring IP Audit for Basic IPS Support, page 24-23

Configuring Threat Detection

This section describes how to configure scanning threat detection and basic threat detection, and also how to use statistics to analyze threats. Threat detection is available in single mode only.

This section includes the following topics:

- Configuring Basic Threat Detection, page 24-1
- Configuring Scanning Threat Detection, page 24-5
- Configuring and Viewing Threat Statistics, page 24-7

Configuring Basic Threat Detection

Basic threat detection detects activity that might be related to an attack, such as a DoS attack. Basic threat detection is enabled by default.

This section includes the following topics:

- Basic Threat Detection Overview, page 24-2
- Configuring Basic Threat Detection, page 24-2
- Managing Basic Threat Statistics, page 24-4

Basic Threat Detection Overview

Using basic threat detection, the security appliance monitors the rate of dropped packets and security events due to the following reasons:

- Denial by access lists
- Bad packet format (such as invalid-ip-header or invalid-tcp-hdr-length)
- Connection limits exceeded (both system-wide resource limits, and limits set in the configuration)
- DoS attack detected (such as an invalid SPI, Stateful Firewall check failure)
- Basic firewall checks failed (This option is a combined rate that includes all firewall-related packet drops in this bulleted list. It does not include non-firewall-related drops such as interface overload, packets failed at application inspection, and scanning attack detected.)
- Suspicious ICMP packets detected
- Packets failed application inspection
- Interface overload
- Scanning attack detected (This option monitors scanning attacks; for example, the first TCP packet is not a SYN packet, or the TCP connection failed the 3-way handshake. Full scanning threat detection (see the "Configuring Scanning Threat Detection" section on page 24-5) takes this scanning attack rate information and acts on it by classifying hosts as attackers and automatically shunning them, for example.)
- Incomplete session detection such as TCP SYN attack detected or no data UDP session attack detected

When the security appliance detects a threat, it immediately sends a system log message (733100).

Basic threat detection affects performance only when there are drops or potential threats; even in this scenario, the performance impact is insignificant.

Configuring Basic Threat Detection

To configure basic threat detection, including enabling or disabling it and changing the default limits, perform the following steps:

Step 1 To enable basic threat detection (if you previously disabled it), enter the following command:

```
hostname(config)# threat-detection basic-threat
```

By default, this command enables detection for certain types of security events, including packet drops and incomplete session detections. You can override the default settings for each type of event if desired.

If an event rate is exceeded, then the security appliance sends a system message. The security appliance tracks two types of rates: the average event rate over an interval, and the burst event rate over a shorter burst interval. The burst rate interval is 1/60th of the average rate interval or 10 seconds, whichever is higher. For each received event, the security appliance checks the average and burst rate limits; if both rates are exceeded, then the security appliance sends two separate system messages, with a maximum of one message for each rate type per burst period.

To disable basic threat detection, enter the **no threat-detection basic-threat** command.

Table 24-1 lists the default settings. You can view all these default settings using the **show running-config all threat-detection** command.

Table 24-1 Basic Threat Detection Default Settings

Packet Drop Reason	Trigger Settings	
	Average Rate	**Burst Rate**
• DoS attack detected • Bad packet format • Connection limits exceeded • Suspicious ICMP packets detected	100 drops/sec over the last 600 seconds.	400 drops/sec over the last 10 second period.
	80 drops/sec over the last 3600 seconds.	320 drops/sec over the last 60 second period.
Scanning attack detected	5 drops/sec over the last 600 seconds.	10 drops/sec over the last 10 second period.
	4 drops/sec over the last 3600 seconds.	8 drops/sec over the last 60 second period.
Incomplete session detected such as TCP SYN attack detected or no data UDP session attack detected (combined)	100 drops/sec over the last 600 seconds.	200 drops/sec over the last 10 second period.
	80 drops/sec over the last 3600 seconds.	160 drops/sec over the last 60 second period.
Denial by access lists	400 drops/sec over the last 600 seconds.	800 drops/sec over the last 10 second period.
	320 drops/sec over the last 3600 seconds.	640 drops/sec over the last 60 second period.
• Basic firewall checks failed • Packets failed application inspection	400 drops/sec over the last 600 seconds.	1600 drops/sec over the last 10 second period.
	320 drops/sec over the last 3600 seconds.	1280 drops/sec over the last 60 second period.
Interface overload	2000 drops/sec over the last 600 seconds.	8000 drops/sec over the last 10 second period.
	1600 drops/sec over the last 3600 seconds.	6400 drops/sec over the last 60 second period.

Step 2 (Optional) To change the default settings for one or more type of event, enter the following command:

```
hostname(config)# threat-detection rate {acl-drop | bad-packet-drop | conn-limit-drop |
dos-drop | fw-drop | icmp-drop | inspect-drop | interface-drop | scanning-threat |
syn-attack} rate-interval rate_interval average-rate av_rate burst-rate burst_rate
```

For a description of each event type, see the "Basic Threat Detection Overview" section on page 24-2.

When you use this command with the **scanning-threat** keyword, it is also used in the scanning threat detection feature (see the "Configuring Scanning Threat Detection" section). The rates you set in this command determine when a host is considered to be an attacker or a target. If you do not set the rates using this command, the default values are used for the scanning threat detection feature as well as the basic threat detection feature. If you do not configure basic threat detection, you can still use this command with the **scanning-threat** keyword to configure the rate limits for scanning threat detection.

The **rate-interface** *rate_interval* argument is between 600 seconds and 2592000 seconds (30 days). The rate interval is used to determine the length of time over which to average the drops. It also determines the burst threshold rate interval (see below).

The **average-rate** *av_rate* argument can be between 0 and 2147483647 in drops/sec.

The **burst-rate** *burst_rate* argument can be between 0 and 2147483647 in drops/sec. The burst rate is calculated as the average rate every *N* seconds, where *N* is the burst rate interval. The burst rate interval is 1/60th of the average rate interval or 10 seconds, whichever is larger.

You can configure up to three different rate intervals for each event type.

The following example enables basic threat detection, and changes the triggers for DoS attacks:

```
hostname(config)# threat-detection basic-threat
hostname(config)# threat-detection rate dos-drop rate-interval 600 average-rate 60
burst-rate 100
```

Managing Basic Threat Statistics

- To view basic threat statistics, enter the following command:

  ```
  hostname# show threat-detection rate [min-display-rate min_display_rate] [acl-drop |
  bad-packet-drop | conn-limit-drop | dos-drop | fw-drop | icmp-drop | inspect-drop |
  interface-drop | scanning-threat | syn-attack]
  ```

  where the **min-display-rate** *min_display_rate* argument limits the display to statistics that exceed the minimum display rate in events per second. You can set the *min_display_rate* between 0 and 2147483647.

  For a description of each event type, see the "Basic Threat Detection Overview" section on page 24-2.

  The output shows the average rate in events/sec over two fixed time periods: the last 10 minutes and the last 1 hour. It also shows: the current burst rate in events/sec over the last completed burst interval, which is 1/60th of the average rate interval or 10 seconds, whichever is larger; the number of times the rates were exceeded (triggered); and the total number of events over the time periods.

  The security appliance stores the count at the end of each burst period, for a total of 60 completed burst intervals. The unfinished burst interval presently occurring is not included in the average rate. For example, if the average rate interval is 20 minutes, then the burst interval is 20 seconds. If the last burst interval was from 3:00:00 to 3:00:20, and you use the **show** command at 3:00:25, then the last 5 seconds are not included in the output.

  The only exception to this rule is if the number of events in the unfinished burst interval already exceeds the number of events in the oldest burst interval (#1 of 60) when calculating the total events. In that case, the security appliance calculates the total events as the last 59 complete intervals, plus the events so far in the unfinished burst interval. This exception lets you monitor a large increase in events in real time.

- To clear basic threat statistics, enter the following command:

  ```
  hostname# clear threat-detection rate
  ```

The following is sample output from the **show threat-detection rate** command:

```
hostname# show threat-detection rate

                        Average(eps)    Current(eps) Trigger         Total events
  10-min ACL  drop:                0              0       0                   16
  1-hour ACL  drop:                0              0       0                  112
  1-hour SYN attck:                5              0       2                21438
  10-min  Scanning:                0              0      29                  193
```

```
1-hour  Scanning:              106             0       10              384776
1-hour Bad  pkts:               76             0       2               274690
10-min  Firewall:                0             0       3                   22
1-hour  Firewall:               76             0       2               274844
10-min DoS attck:                0             0       0                    6
1-hour DoS attck:                0             0       0                   42
10-min Interface:                0             0       0                  204
1-hour Interface:               88             0       0               318225
```

Configuring Scanning Threat Detection

A typical scanning attack consists of a host that tests the accessibility of every IP address in a subnet (by scanning through many hosts in the subnet or sweeping through many ports in a host or subnet). The scanning threat detection feature determines when a host is performing a scan. Unlike IPS scan detection that is based on traffic signatures, the security appliance scanning threat detection feature maintains an extensive database that contains host statistics that can be analyzed for scanning activity.

The host database tracks suspicious activity such as connections with no return activity, access of closed service ports, vulnerable TCP behaviors such as non-random IPID, and many more behaviors.

You can configure the security appliance to send system log messages about an attacker or you can automatically shun the host.

Caution The scanning threat detection feature can affect the security appliance performance and memory significantly while it creates and gathers host- and subnet-based data structure and information.

This section includes the following topics:

- Enabling Scanning Threat Detection, page 24-5
- Managing Shunned Hosts, page 24-6
- Viewing Attackers and Targets, page 24-7

Enabling Scanning Threat Detection

To configure scanning threat detection, perform the following steps:

Step 1 To enable scanning threat detection, enter the following command:

```
hostname(config)# threat-detection scanning-threat [shun
[except {ip-address ip_address mask | object-group network_object_group_id}]]
```

By default, the system log message 730101 is generated when a host is identified as an attacker.

The **shun** keyword automatically terminates a host connection when the security appliance identifies the host as an attacker, in addition to sending the system log message.

You can except host IP addresses from being shunned by entering the **except ip-address** or **except object-group** keywords. Enter this command multiple times to identify multiple IP addresses or network object groups to exempt from shunning.

Step 2 (Optional) To set the duration of the shun for attacking hosts, enter the following command:

```
hostname(config)# threat-detection scanning-threat shun duration seconds
```

where the *seconds* argument is between 10 and 2592000 seconds. The default is 3600 seconds (1 hour).

Step 3 (Optional) To change the default event limit for when the security appliance identifies a host as an attacker or as a target, enter the following command:

```
hostname(config)# threat-detection rate scanning-threat rate-interval rate_interval
average-rate av_rate burst-rate burst_rate
```

If the scanning threat rate is exceeded, then the security appliance sends a system message, and optionally shuns the attacker. The security appliance tracks two types of rates: the average event rate over an interval, and the burst event rate over a shorter burst interval. The burst event rate is 1/60th of the average rate interval or 10 seconds, whichever is higher. For each event detected that is considered to be part of a scanning attack, the security appliance checks the average and burst rate limits. If either rate is exceeded for traffic sent from a host, then that host is considered to be an attacker. If either rate is exceeded for traffic received by a host, then that host is considered to be a target.

If you already configured this command as part of the basic threat detection configuration (see the "Configuring Basic Threat Detection" section on page 24-1), then those settings are shared with the scanning threat detection feature; you cannot configure separate rates for each feature. If you do not set the rates using this command, the default values are used for both the scanning threat detection feature and the basic threat detection feature. The default values are:

Table 24-2 Default Rate Limits for Scanning Threat Detection

Average Rate	Burst Rate
5 drops/sec over the last 600 seconds.	10 drops/sec over the last 10 second period.
5 drops/sec over the last 3600 seconds.	10 drops/sec over the last 60 second period.

The *rate_interval* is between 300 seconds and 2592000 seconds (30 days). The rate interval is used to determine the length of time over which to average the events. It also determines the burst threshold rate interval (see below).

The **average-rate** *av_rate* argument can be between 0 and 2147483647 in drops/sec.

The **burst-rate** *burst_rate* argument can be between 0 and 2147483647 in drops/sec. The burst rate is calculated as the average rate every *N* seconds, where *N* is the burst rate interval. The burst rate interval is 1/60th of the rate interval or 10 seconds, whichever is larger.

You can configure up to three commands with different rate intervals.

The following example enables scanning threat detection and automatically shuns hosts categorized as attackers, except for hosts on the 10.1.1.0 network. The default rate limits for scanning threat detection are also changed.

```
hostname(config)# threat-detection scanning-threat shun except ip-address 10.1.1.0
255.255.255.0
hostname(config)# threat-detection rate scanning-threat rate-interval 1200 average-rate 10
burst-rate 20
hostname(config)# threat-detection rate scanning-threat rate-interval 2400 average-rate 10
burst-rate 20
```

Managing Shunned Hosts

- To view the hosts that are currently shunned, enter the following command:

  ```
  hostname# show threat-detection shun
  ```

- To release a host from being shunned, enter the following command:

  ```
  hostname# clear threat-detection shun [ip_address [mask]]
  ```

  If you do not specify an IP address, all hosts are cleared from the shun list.

The following is sample output from the **show threat-detection shun** command:

```
hostname# show threat-detection shun
Shunned Host List:
10.1.1.6
192.168.6.7
```

Viewing Attackers and Targets

To view the hosts that the security appliance decides are attackers (including hosts on the shun list), and to view the hosts that are the target of an attack, enter the following command:

```
hostname# show threat-detection scanning-threat [attacker | target]
```

If you do not enter an option, both attackers and target hosts are displayed.

The following is sample output from the **show threat-detection scanning-threat attacker** command:

```
hostname# show threat-detection scanning-threat attacker
10.1.2.3
10.8.3.6
209.165.200.225
```

Configuring and Viewing Threat Statistics

You can configure the security appliance to collect extensive statistics. Threat detection statistics show both allowed and dropped traffic rates. To view statistics for basic threat detection, see the "Managing Basic Threat Statistics" section on page 24-4. By default, statistics for access lists are enabled.

Caution Enabling statistics can affect the security appliance performance, depending on the type of statistics enabled. The **threat-detection statistics host** command affects performance in a significant way; if you have a high traffic load, you might consider enabling this type of statistics temporarily. The **threat-detection statistics port** command, however, has modest impact.

This section includes the following topics:

- Configuring Threat Statistics, page 24-7
- Viewing Threat Statistics, page 24-8

Configuring Threat Statistics

By default, statistics for access lists are enabled. To enable *all* statistics, enter the following command:

```
hostname(config)# threat-detection statistics
```

To enable only certain statistics, enter one or more of the following commands for each statistic type.

- Access lists—To enable statistics for access lists (if they were disabled previously), enter the following command:

```
hostname(config)# threat-detection statistics access-list
```

Access list statistics are only displayed using the **show threat-detection top access-list** command.

- Hosts—To enable statistics for hosts, enter the following command:

  ```
  hostname(config)# threat-detection statistics host
  ```

  The host statistsics accumulate for as long as the host is active and in the scanning threat host database. The host is deleted from the database (and the statistics cleared) after 10 minutes of inactivity.

- TCP and UDP ports—To enable statistics for TCP and UDP ports, enter the following command:

  ```
  hostname(config)# threat-detection statistics port
  ```

- Non-TCP/UDP IP ports—To enable statistics for non-TCP/UDP IP protocols, enter the following command:

  ```
  hostname(config)# threat-detection statistics protocol
  ```

- TCP Intercept—To enable statistics for attacks intercepted by TCP Intercept (see the "Configuring Connection Limits and Timeouts" section on page 24-17 to enable TCP Intercept), enter the following command:

  ```
  hostname(config)# threat-detection statistics tcp-intercept [rate-interval minutes]
  [burst-rate attacks_per_sec] [average-rate attacks_per_sec]
  ```

  where the **rate-interval** *minutes* argument sets the size of the history monitoring window, between 1 and 1440 minutes. The default is 30 minutes. The security appliance samples the number of attacks 60 times during the rate interval, so for the default 30 minute period, statistics are collected every 60 seconds.

  The **burst-rate** *attacks_per_sec* argument sets the threshold for syslog message generation, between 25 and 2147483647. The default is 400 per second. When the burst rate is exceeded, syslog message 733104 is generated.

  The **average-rate** *attacks_per_sec* argument sets the average rate threshold for syslog message generation, between 25 and 2147483647. The default is 200 per second. When the average rate is exceeded, syslog message 733105 is generated.

Viewing Threat Statistics

The display output shows the following:

- The average rate in events/sec over fixed time periods.
- The current burst rate in events/sec over the last completed burst interval, which is 1/60th of the average rate interval or 10 seconds, whichever is larger
- The number of times the rates were exceeded (for dropped traffic statistics only)
- The total number of events over the fixed time periods.

The security appliance stores the count at the end of each burst period, for a total of 60 completed burst intervals. The unfinished burst interval presently occurring is not included in the average rate. For example, if the average rate interval is 20 minutes, then the burst interval is 20 seconds. If the last burst interval was from 3:00:00 to 3:00:20, and you use the **show** command at 3:00:25, then the last 5 seconds are not included in the output.

The only exception to this rule is if the number of events in the unfinished burst interval already exceeds the number of events in the oldest burst interval (#1 of 60) when calculating the total events. In that case, the security appliance calculates the total events as the last 59 complete intervals, plus the events so far in the unfinished burst interval. This exception lets you monitor a large increase in events in real time.

To view statistics, enter one of the following commands.

- To view the top 10 statistics, enter the following command:

  ```
  hostname# show threat-detection statistics [min-display-rate min_display_rate] top
  [[access-list | host | port-protocol] [rate-1 | rate-2 | rate-3] |
  tcp-intercept [all] detail]]
  ```

  where the **min-display-rate** *min_display_rate* argument limits the display to statistics that exceed the minimum display rate in events per second. You can set the *min_display_rate* between 0 and 2147483647.

  If you do not enter any options, the top 10 statistics are shown for all categories.

  To view the top 10 ACEs that match packets, including both permit and deny ACEs., use the **access-list** keyword. Permitted and denied traffic are not differentiated in this display. If you enable basic threat detection using the **threat-detection basic-threat** command, you can track access list denies using the **show threat-detection rate access-list** command.

  To view only host statistics, use the **host** keyword.

  To view statistics for ports and protocols, use the **port-protocol** keyword. The **port-protocol** keyword shows the combined statistics of TCP/UDP port and IP protocol types. TCP (protocol 6) and UDP (protocol 17) are not included in the display for IP protocols; TCP and UDP ports are, however, included in the display for ports. If you only enable statistics for one of these types, port or protocol, then you will only view the enabled statistics.

  To view TCP Intercept statistics, use the **tcp-intercept** keyword. The display includes the top 10 protected servers under attack. The **all** keyword to shows the history data of all the traced servers. The **detail** keyword shows history sampling data. The security appliance samples the number of attacks 60 times during the rate interval, so for the default 30 minute period, statistics are collected every 60 seconds.

  The **rate-1** keyword shows the statistics for the smallest fixed rate intervals available in the display; **rate-2** shows the next largest rate interval; and **rate-3**, if you have three intervals defined, shows the largest rate interval. For example, the display shows statistics for the last 1 hour, 8 hours, and 24 hours. If you set the **rate-1** keyword, the security appliance shows only the 1 hour time interval.

- To view statistics for all hosts or for a specific host or subnet, enter the following command:

  ```
  hostname# show threat-detection statistics [min-display-rate min_display_rate] host
  [ip_address [mask]]
  ```

- To view statistics for all ports or for a specific port or range of ports, enter the following command:

  ```
  hostname# show threat-detection statistics [min-display-rate min_display_rate] port
  [start_port[-end_port]]
  ```

- To view statistics for all IP protocols or for a specific protocol, enter the following command:

  ```
  hostname# show threat-detection statistics [min-display-rate min_display_rate]
  protocol [protocol_number | ah | eigrp | esp | gre | icmp | igmp | igrp | ip | ipinip
  | ipsec | nos | ospf | pcp | pim | pptp | snp | tcp | udp]
  ```

  where the *protocol_number* argument is an integer between 0 and 255.

The following is sample output from the **show threat-detection statistics host** command:

```
hostname# show threat-detection statistics host
```

```
                          Average(eps)    Current(eps) Trigger         Total events
Host:10.0.0.1: tot-ses:289235 act-ses:22571 fw-drop:0 insp-drop:0 null-ses:21438 bad-acc:0
  1-hour Sent byte:               2938               0       0             10580308
  8-hour Sent byte:                367               0       0             10580308
 24-hour Sent byte:                122               0       0             10580308
  1-hour Sent pkts:                 28               0       0               104043
  8-hour Sent pkts:                  3               0       0               104043
 24-hour Sent pkts:                  1               0       0               104043
  20-min Sent drop:                  9               0       1                10851
  1-hour Sent drop:                  3               0       1                10851
  1-hour Recv byte:               2697               0       0              9712670
  8-hour Recv byte:                337               0       0              9712670
 24-hour Recv byte:                112               0       0              9712670
  1-hour Recv pkts:                 29               0       0               104846
  8-hour Recv pkts:                  3               0       0               104846
 24-hour Recv pkts:                  1               0       0               104846
  20-min Recv drop:                 42               0       3                50567
  1-hour Recv drop:                 14               0       1                50567
Host:10.0.0.0: tot-ses:1 act-ses:0 fw-drop:0 insp-drop:0 null-ses:0 bad-acc:0
  1-hour Sent byte:                  0               0       0                  614
  8-hour Sent byte:                  0               0       0                  614
 24-hour Sent byte:                  0               0       0                  614
  1-hour Sent pkts:                  0               0       0                    6
  8-hour Sent pkts:                  0               0       0                    6
 24-hour Sent pkts:                  0               0       0                    6
  20-min Sent drop:                  0               0       0                    4
  1-hour Sent drop:                  0               0       0                    4
  1-hour Recv byte:                  0               0       0                  706
  8-hour Recv byte:                  0               0       0                  706
 24-hour Recv byte:                  0               0       0                  706
  1-hour Recv pkts:                  0               0       0                    7
```

Table 24-3 shows each field description.

Table 24-3 show threat-detection statistics host Fields

Field	Description
Host	Shows the host IP address.
tot-ses	Shows the total number of sessions for this host since it was added to the database.
act-ses	Shows the total number of active sessions that the host is currently involved in.
fw-drop	Shows the number of firewall drops. Firewall drops is a combined rate that includes all firewall-related packet drops tracked in basic threat detection, including access list denials, bad packets, exceeded connection limits, DoS attack packets, suspicious ICMP packets, TCP SYN attack packets, and no data UDP attack packets. It does not include non-firewall-related drops such as interface overload, packets failed at application inspection, and scanning attack detected.
insp-drop	Shows the number of packets dropped because they failed application inspection.
null-ses	Shows the number of null sessions, which are TCP SYN sessions that did not complete within the 3-second timeout, and UDP sessions that did not have any data sent by its server 3 seconds after the session starts.

Table 24-3 show threat-detection statistics host Fields (continued)

Field	Description
bad-acc	Shows the number of bad access attempts to host ports that are in a closed state. When a port is determined to be in a null session (see above), the port state of the host is set to HOST_PORT_CLOSE. Any client accessing the port of the host is immediately classified as a bad access without the need to wait for a timeout.
Average(eps)	Shows the average rate in events/sec over each time period. The security appliance stores the count at the end of each burst period, for a total of 60 completed burst intervals. The unfinished burst interval presently occurring is not included in the average rate. For example, if the average rate interval is 20 minutes, then the burst interval is 20 seconds. If the last burst interval was from 3:00:00 to 3:00:20, and you use the **show** command at 3:00:25, then the last 5 seconds are not included in the output. The only exception to this rule is if the number of events in the unfinished burst interval already exceeds the number of events in the oldest burst interval (#1 of 60) when calculating the total events. In that case, the security appliance calculates the total events as the last 59 complete intervals, plus the events so far in the unfinished burst interval. This exception lets you monitor a large increase in events in real time.
Current(eps)	Shows the current burst rate in events/sec over the last completed burst interval, which is 1/60th of the average rate interval or 10 seconds, whichever is larger. For the example specified in the Average(eps) description, the current rate is the rate from 3:19:30 to 3:20:00
Trigger	Shows the number of times the dropped packet rate limits were exceeded. For valid traffic identified in the sent and received bytes and packets rows, this value is always 0, because there are no rate limits to trigger for valid traffic.
Total events	Shows the total number of events over each rate interval. The unfinished burst interval presently occurring is not included in the total events. The only exception to this rule is if the number of events in the unfinished burst interval already exceeds the number of events in the oldest burst interval (#1 of 60) when calculating the total events. In that case, the security appliance calculates the total events as the last 59 complete intervals, plus the events so far in the unfinished burst interval. This exception lets you monitor a large increase in events in real time.
20-min, 1-hour, 8-hour, and 24-hour	Shows statistics for these fixed rate intervals.
Sent byte	Shows the number of successful bytes sent from the host.
Sent pkts	Shows the number of successful packets sent from the host.
Sent drop	Shows the number of packets sent from the host that were dropped because they were part of a scanning attack.
Recv byte	Shows the number of successful bytes received by the host.
Recv pkts	Shows the number of successful packets received by the host.
Recv drop	Shows the number of packets received by the host that were dropped because they were part of a scanning attack.

Configuring TCP Normalization

The TCP normalization feature identifies abnormal packets that the security appliance can act on when they are detected; for example, the security appliance can allow, drop, or clear the packets. TCP normalization helps protect the security appliance from attacks. This section includes the following topics:

- TCP Normalization Overview, page 24-12
- Enabling the TCP Normalizer, page 24-12

TCP Normalization Overview

The TCP normalizer includes non-configurable actions and configurable actions. Typically, non-configurable actions that drop or clear connections apply to packets that are always bad. Configurable actions (as detailed in "Enabling the TCP Normalizer" section on page 24-12) might need to be customized depending on your network needs.

See the following guidelines for TCP normalization:

- The normalizer does not protect from SYN floods. The security appliance includes SYN flood protection in other ways.
- The normalizer always sees the SYN packet as the first packet in a flow unless the security appliance is in loose mode due to failover.

Enabling the TCP Normalizer

This feature uses Modular Policy Framework, so that implementing TCP normalization consists of identifying traffic, specifying the TCP normalization actions, and activating TCP normalization on an interface. See Chapter 16, "Using Modular Policy Framework," for more information.

To configure TCP normalization, perform the following steps:

Step 1 To specify the TCP normalization criteria that you want to look for, create a TCP map by entering the following command:

```
hostname(config)# tcp-map tcp-map-name
```

For each TCP map, you can customize one or more settings.

Step 2 (Optional) Configure the TCP map criteria by entering one or more of the following commands (see Table 24-4). If you want to use the default settings for all criteria, you do not need to enter any commands for the TCP map. If you want to customize some settings, then the defaults are used for any commands you do not enter. The default configuration includes the following settings:

```
no check-retransmission
no checksum-verification
exceed-mss allow
queue-limit 0 timeout 4
reserved-bits allow
syn-data allow
synack-data drop
invalid-ack drop
seq-past-window drop
tcp-options range 6 7 clear
tcp-options range 9 255 clear
```

```
tcp-options selective-ack allow
tcp-options timestamp allow
tcp-options window-scale allow
ttl-evasion-protection
urgent-flag clear
window-variation allow-connection
```

Table 24-4 tcp-map Commands

Command	Notes
check-retransmission	Prevents inconsistent TCP retransmissions.
checksum-verification	Verifies the checksum.
exceed-mss {**allow** | **drop**}	Sets the action for packets whose data length exceeds the TCP maximum segment size. (Default) The **allow** keyword allows packets whose data length exceeds the TCP maximum segment size. The **drop** keyword drops packets whose data length exceeds the TCP maximum segment size.
invalid-ack {**allow** | **drop**}	Sets the action for packets with an invalid ACK. You might see invalid ACKs in the following instances: • In the TCP connection SYN-ACK-received status, if the ACK number of a received TCP packet is not exactly same as the sequence number of the next TCP packet sending out, it is an invalid ACK. • Whenever the ACK number of a received TCP packet is greater than the sequence number of the next TCP packet sending out, it is an invalid ACK. The **allow** keyword allows packets with an invalid ACK. (Default) The **drop** keyword drops packets with an invalid ACK. **Note** TCP packets with an invalid ACK are automatically allowed for WAAS connections.

Table 24-4 tcp-map Commands (continued)

Command	Notes
queue-limit *pkt_num* [**timeout** *seconds*]	Sets the maximum number of out-of-order packets that can be buffered and put in order for a TCP connection, between 1 and 250 packets. The default is 0, which means this setting is disabled and the default system queue limit is used depending on the type of traffic: • Connections for application inspection (the **inspect** command), IPS (the **ips** command), and TCP check-retransmission (the TCP map **check-retransmission** command) have a queue limit of 3 packets. If the security appliance receives a TCP packet with a different window size, then the queue limit is dynamically changed to match the advertised setting. • For other TCP connections, out-of-order packets are passed through untouched. If you set the **queue-limit** command to be 1 or above, then the number of out-of-order packets allowed for all TCP traffic matches this setting. For application inspection, IPS, and TCP check-retransmission traffic, any advertised settings are ignored. For other TCP traffic, out-of-order packets are now buffered and put in order instead of passed through untouched. The **timeout** *seconds* argument sets the maximum amount of time that out-of-order packets can remain in the buffer, between 1 and 20 seconds; if they are not put in order and passed on within the timeout period, then they are dropped. The default is 4 seconds. You cannot change the timeout for any traffic if the *pkt_num* argument is set to 0; you need to set the limit to be 1 or above for the **timeout** keyword to take effect.
reserved-bits {**allow** | **clear** | **drop**}	Sets the action for reserved bits in the TCP header. (Default) The **allow** keyword allows packets with the reserved bits in the TCP header. The **clear** keyword clears the reserved bits in the TCP header and allows the packet. The **drop** keyword drops the packet with the reserved bits in the TCP header.
seq-past-window {**allow** | **drop**}	Sets the action for packets that have past-window sequence numbers, namely the sequence number of a received TCP packet is greater than the right edge of the TCP receiving window. The **allow** keyword allows packets that have past-window sequence numbers. This action is only allowed if the **queue-limit** command is set to 0 (disabled). (Default) The **drop** keyword drops packets that have past-window sequence numbers.

Table 24-4 tcp-map Commands (continued)

Command	Notes
synack-data {**allow** | **drop**}	Sets the action for TCP SYNACK packets that contain data. The **allow** keyword allows TCP SYNACK packets that contain data. (Default) The **drop** keyword drops TCP SYNACK packets that contain data.
syn-data {**allow** | **drop**}	Sets the action for SYN packets with data. (Default) The **allow** keyword allows SYN packets with data. The **drop** keyword drops SYN packets with data.
tcp-options {**selective-ack** | **timestamp** | **window-scale**} {**allow** | **clear**} Or **tcp-options range** *lower upper* {**allow** | **clear** | **drop**}	Sets the action for packets with TCP options, including the selective-ack, timestamp, or window-scale TCP options. (Default) The **allow** keyword allows packets with the specified option. (Default for **range**) The **clear** keyword clears the option and allows the packet. The **drop** keyword drops the packet with the specified option. The **selective-ack** keyword sets the action for the SACK option. The **timestamp** keyword sets the action for the timestamp option. Clearing the timestamp option disables PAWS and RTT. The **widow-scale** keyword sets the action for the window scale mechanism option. The **range** keyword specifies a range of options. The *lower* argument sets the lower end of the range as 6, 7, or 9 through 255. The *upper* argument sets the upper end of the range as 6, 7, or 9 through 255.
ttl-evasion-protection	Disables the TTL evasion protection. Do not enter this command it you want to prevent attacks that attempt to evade security policy. For example, an attacker can send a packet that passes policy with a very short TTL. When the TTL goes to zero, a router between the security appliance and the endpoint drops the packet. It is at this point that the attacker can send a malicious packet with a long TTL that appears to the security appliance to be a retransmission and is passed. To the endpoint host, however, it is the first packet that has been received by the attacker. In this case, an attacker is able to succeed without security preventing the attack.

Table 24-4 tcp-map Commands (continued)

Command	Notes
urgent-flag {**allow** | **clear**}	Sets the action for packets with the URG flag. The URG flag is used to indicate that the packet contains information that is of higher priority than other data within the stream. The TCP RFC is vague about the exact interpretation of the URG flag, therefore end systems handle urgent offsets in different ways, which may make the end system vulnerable to attacks. The **allow** keyword allows packets with the URG flag. (Default) The **clear** keyword clears the URG flag and allows the packet.
window-variation {**allow** | **drop**}	Sets the action for a connection that has changed its window size unexpectedly. The window size mechanism allows TCP to advertise a large window and to subsequently advertise a much smaller window without having accepted too much data. From the TCP specification, "shrinking the window" is strongly discouraged. When this condition is detected, the connection can be dropped. (Default) The **allow** keyword allows connections with a window variation. The **drop** keyword drops connections with a window variation.

Step 3 To identify the traffic, add a class map using the **class-map** command. See the "Creating a Layer 3/4 Class Map for Through Traffic" section on page 16-5 for more information.

For example, you can match all traffic using the following commands:

```
hostname(config)# class-map TCPNORM
hostname(config-cmap)# match any
```

To match specific traffic, you can match an access list:

```
hostname(config)# access list TCPNORM extended permit ip any 10.1.1.1 255.255.255.255
hostname(config)# class-map TCP_norm_class
hostname(config-cmap)# match access-list TCPNORM
```

Step 4 To add or edit a policy map that sets the actions to take with the class map traffic, enter the following commands:

```
hostname(config)# policy-map name
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

where the *class_map_name* is the class map from Step 1.

For example:

```
hostname(config)# policy-map TCP_norm_policy
hostname(config-pmap)# class TCP_norm_class
hostname(config-pmap-c)#
```

Step 5 Apply the TCP map to the class map by entering the following command.

```
hostname(config-pmap-c)# set connection advanced-options tcp-map-name
```

Step 6 To activate the policy map on one or more interfaces, enter the following command:

```
hostname(config)# service-policy policymap_name {global | interface interface_name}
```

Where **global** applies the policy map to all interfaces, and **interface** applies the policy to one interface. Only one global policy is allowed. Interface service policies take precedence over the global service policy for a given feature. For example, if you have a global policy with inspections, and an interface policy with TCP normalization, then both inspections and TCP normalization are applied to the interface. However, if you have a global policy with inspections, and an interface policy with inspections, then only the interface policy inspections are applied to that interface.

For example, to allow urgent flag and urgent offset packets for all traffic sent to the range of TCP ports between the well known FTP data port and the Telnet port, enter the following commands:

```
hostname(config)# tcp-map tmap
hostname(config-tcp-map)# urgent-flag allow
hostname(config-tcp-map)# class-map urg-class
hostname(config-cmap)# match port tcp range ftp-data telnet
hostname(config-cmap)# policy-map pmap
hostname(config-pmap)# class urg-class
hostname(config-pmap-c)# set connection advanced-options tmap
hostname(config-pmap-c)# service-policy pmap global
```

Configuring Connection Limits and Timeouts

This section describes how to set maximum TCP and UDP connections, maximum embryonic connections, maximum per-client connections, connection timeouts, dead connection detection, and how to disable TCP sequence randomization. You can set limits for connections that go through the security appliance, or for management connections to the security appliance. This section contains the following topics:

- Connection Limit Overview, page 24-17
- Enabling Connection Limits and Timeouts, page 24-19

Note You can also configure maximum connections, maximum embryonic connections, and TCP sequence randomization in the NAT configuration. If you configure these settings for the same traffic using both methods, then the security appliance uses the lower limit. For TCP sequence randomization, if it is disabled using either method, then the security appliance disables TCP sequence randomization.

Connection Limit Overview

This section describes why you might want to limit connections, and includes the following topics:

- TCP Intercept Overview, page 24-18
- Disabling TCP Intercept for Management Packets for Clientless SSL Compatibility, page 24-18
- Dead Connection Detection (DCD) Overview, page 24-18
- TCP Sequence Randomization Overview, page 24-18

TCP Intercept Overview

Limiting the number of embryonic connections protects you from a DoS attack. The security appliance uses the per-client limits and the embryonic connection limit to trigger TCP Intercept, which protects inside systems from a DoS attack perpetrated by flooding an interface with TCP SYN packets. An embryonic connection is a connection request that has not finished the necessary handshake between source and destination. TCP Intercept uses the SYN cookies algorithm to prevent TCP SYN-flooding attacks. A SYN-flooding attack consists of a series of SYN packets usually originating from spoofed IP addresses. The constant flood of SYN packets keeps the server SYN queue full, which prevents it from servicing connection requests. When the embryonic connection threshold of a connection is crossed, the security appliance acts as a proxy for the server and generates a SYN-ACK response to the client SYN request. When the security appliance receives an ACK back from the client, it can then authenticate the client and allow the connection to the server.

To view TCP Intercept statistics, including the top 10 servers under attack, see the "Configuring and Viewing Threat Statistics" section on page 24-7.

Disabling TCP Intercept for Management Packets for Clientless SSL Compatibility

By default, TCP management connections have TCP Intercept always enabled. When TCP Intercept is enabled, it intercepts the 3-way TCP connection establishment handshake packets and thus deprives the security appliance from processing the packets for clientless SSL. Clientless SSL requires the ability to process the 3-way handshake packets to provide selective ACK and other TCP options for clientless SSL connections. To disable TCP Intercept for management traffic, you can set the embryonic connection limit; only after the embryonic connection limit is reached is TCP Intercept enabled.

Dead Connection Detection (DCD) Overview

DCD detects a dead connection and allows it to expire, without expiring connections that can still handle traffic. You configure DCD when you want idle, but valid connections to persist.

When you enable DCD, idle timeout behavior changes. With idle timeout, DCD probes are sent to each of the two end-hosts to determine the validity of the connection. If an end-host fails to respond after probes are sent at the configured intervals, the connection is freed, and reset values, if configured, are sent to each of the end-hosts. If both end-hosts respond that the connection is valid, the activity timeout is updated to the current time and the idle timeout is rescheduled accordingly.

Enabling DCD changes the behavior of idle-timeout handling in the TCP normalizer. DCD probing resets the idle timeout on the connections seen in the **show conn** command. To determine when a connection that has exceeded the configured timeout value in the timeout command but is kept alive due to DCD probing, the **show service-policy** command includes counters to show the amount of activity from DCD.

TCP Sequence Randomization Overview

Each TCP connection has two ISNs: one generated by the client and one generated by the server. The security appliance randomizes the ISN of the TCP SYN passing in both the inbound and outbound directions.

Randomizing the ISN of the protected host prevents an attacker from predecting the next ISN for a new connection and potentially hijacking the new session.

TCP initial sequence number randomization can be disabled if required. For example:

- If another in-line firewall is also randomizing the initial sequence numbers, there is no need for both firewalls to be performing this action, even though this action does not affect the traffic.
- If you use eBGP multi-hop through the security appliance, and the eBGP peers are using MD5. Randomization breaks the MD5 checksum.
- You use a WAAS device that requires the security appliance not to randomize the sequence numbers of connections.

Enabling Connection Limits and Timeouts

To set connection limits and timeouts, perform the following steps:

Step 1 To identify the traffic, add a class map using the **class-map** command. See the "Creating a Layer 3/4 Class Map for Through Traffic" section on page 16-5 or the "Creating a Layer 3/4 Class Map for Management Traffic" section on page 16-7 for more information.

For example, you can match all traffic using the following commands:

```
hostname(config)# class-map CONNS
hostname(config-cmap)# match any
```

To match specific traffic, you can match an access list:

```
hostname(config)# access list CONNS extended permit ip any 10.1.1.1 255.255.255.255
hostname(config)# class-map CONNS
hostname(config-cmap)# match access-list CONNS
```

Step 2 To add or edit a policy map that sets the actions to take with the class map traffic, enter the following commands:

```
hostname(config)# policy-map name
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

where the *class_map_name* is the class map from Step 1.

For example:

```
hostname(config)# policy-map CONNS
hostname(config-pmap)# class CONNS
hostname(config-pmap-c)#
```

Step 3 To set maximum connection limits or whether TCP sequence randomization is enabled, enter the following command:

```
hostname(config-pmap-c)# set connection {[conn-max n] [embryonic-conn-max n]
[per-client-embryonic-max n] [per-client-max n] [random-sequence-number {enable |
disable}]}
```

where the **conn-max** *n* argument sets the maximum number of simultaneous TCP and/or UDP connections that are allowed, between 0 and 65535. The default is 0, which allows unlimited connections.

The **embryonic-conn-max** *n* argument sets the maximum number of simultaneous embryonic connections allowed, between 0 and 65535. The default is 0, which allows unlimited connections.

The **per-client-embryonic-max** *n* argument sets the maximum number of simultaneous embryonic connections allowed per client, between 0 and 65535. The default is 0, which allows unlimited connections.

The **per-client-max** *n* argument sets the maximum number of simultaneous connections allowed per client, between 0 and 65535. The default is 0, which allows unlimited connections.

The **random-sequence-number** {**enable** | **disable**} keyword enables or disables TCP sequence number randomization. See the "TCP Sequence Randomization Overview" section on page 24-18 section for more information.

You can enter this command all on one line (in any order), or you can enter each attribute as a separate command. The security appliance combines the command into one line in the running configuration.

Note For management traffic, you can only set the **conn-max** and **embryonic-conn-max** keywords.

Step 4 To set connection timeouts, enter the following command:

```
hostname(config-pmap-c)# set connection timeout {[embryonic hh:mm:ss] {tcp hh:mm:ss
[reset]] [half-closed hh:mm:ss] [dcd hh:mm:ss [max_retries]]}
```

where the **embryonic** *hh*:*mm*:*ss* keyword sets the timeout period until a TCP embryonic (half-open) connection is closed, between 0:0:5 and 1193:00:00. The default is 0:0:30. You can also set this value to 0, which means the connection never times out.

The **tcp** *hh*:*mm*:*ss* keyword sets the idle timeout between 0:5:0 and 1193:00:00. The default is 1:0:0. You can also set this value to 0, which means the connection never times out. The **reset** keyword sends a reset to TCP endpoints when the connection times out. The security appliance sends the reset packet only in response to a host sending another packet for the timed-out flow (on the same source and destination port). The host then removes the connection from its connection table after receiving the reset packet. The host application can then attempt to establish a new connection using a SYN packet.

The **half-closed** *hh*:*mm*:*ss* keyword sets the idle timeout between 0:5:0 and 1193:00:00. The default is 0:10:0. Half-closed connections are not affected by DCD. Also, the security appliance does not send a reset when taking down half-closed connections.

The **dcd** keyword enables DCD. DCD detects a dead connection and allows it to expire, without expiring connections that can still handle traffic. You configure DCD when you want idle, but valid connections to persist. After a TCP connection times out, the security appliance sends DCD probes to the end hosts to determine the validity of the connection. If one of the end hosts fails to respond after the maximum retries are exhausted, the security appliance frees the connection. If both end hosts respond that the connection is valid, the security appliance updates the activity timeout to the current time and reschedules the idle timeout accordingly. The *retry-interval* sets the time duration in *hh*:*mm*:*ss* format to wait after each unresponsive DCD probe before sending another probe, between 0:0:1 and 24:0:0. The default is 0:0:15. The *max-retries* sets the number of consecutive failed retries for DCD before declaring the connection as dead. The minimum value is 1 and the maximum value is 255. The default is 5.

You can enter this command all on one line (in any order), or you can enter each attribute as a separate command. The command is combined onto one line in the running configuration.

Note This command is not available for management traffic.

Step 5 To activate the policy map on one or more interfaces, enter the following command:

```
hostname(config)# service-policy policymap_name {global | interface interface_name}
```

where *policy_map_name* is the policy map you configured in Step 2. To apply the policy map to traffic on all the interfaces, use the **global** keyword. To apply the policy map to traffic on a specific interface, use the **interface** *interface_name* option, where *interface_name* is the name assigned to the interface with the **nameif** command.

Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.

The following example sets the connection limits and timeouts for all traffic:

```
hostname(config)# class-map CONNS
hostname(config-cmap)# match any
hostname(config-cmap)# policy-map CONNS
hostname(config-pmap)# class CONNS
hostname(config-pmap-c)# set connection conn-max 1000 embryonic-conn-max 3000
hostname(config-pmap-c)# set connection timeout tcp 2:0:0 embryonic 0:40:0 half-closed
0:20:0 dcd
hostname(config-pmap-c)# service-policy CONNS interface outside
```

You can enter **set connection** commands with multiple parameters or you can enter each parameter as a separate command. The security appliance combines the commands into one line in the running configuration. For example, if you entered the following two commands in class configuration mode:

```
hostname(config-pmap-c)# set connection conn-max 600
hostname(config-pmap-c)# set connection embryonic-conn-max 50
```

the output of the **show running-config policy-map** command would display the result of the two commands in a single, combined command:

```
set connection conn-max 600 embryonic-conn-max 50
```

Preventing IP Spoofing

This section lets you enable Unicast Reverse Path Forwarding on an interface. Unicast RPF guards against IP spoofing (a packet uses an incorrect source IP address to obscure its true source) by ensuring that all packets have a source IP address that matches the correct source interface according to the routing table.

Normally, the security appliance only looks at the destination address when determining where to forward the packet. Unicast RPF instructs the security appliance to also look at the source address; this is why it is called Reverse Path Forwarding. For any traffic that you want to allow through the security appliance, the security appliance routing table must include a route back to the source address. See RFC 2267 for more information.

For outside traffic, for example, the security appliance can use the default route to satisfy the Unicast RPF protection. If traffic enters from an outside interface, and the source address is not known to the routing table, the security appliance uses the default route to correctly identify the outside interface as the source interface.

If traffic enters the outside interface from an address that is known to the routing table, but is associated with the inside interface, then the security appliance drops the packet. Similarly, if traffic enters the inside interface from an unknown source address, the security appliance drops the packet because the matching route (the default route) indicates the outside interface.

Unicast RPF is implemented as follows:

- ICMP packets have no session, so each packet is checked.
- UDP and TCP have sessions, so the initial packet requires a reverse route lookup. Subsequent packets arriving during the session are checked using an existing state maintained as part of the session. Non-initial packets are checked to ensure they arrived on the same interface used by the initial packet.

To enable Unicast RPF, enter the following command:

```
hostname(config)# ip verify reverse-path interface interface_name
```

Configuring the Fragment Size

By default, the security appliance allows up to 24 fragments per IP packet, and up to 200 fragments awaiting reassembly. You might need to let fragments on your network if you have an application that routinely fragments packets, such as NFS over UDP. However, if you do not have an application that fragments traffic, we recommend that you do not allow fragments through the security appliance. Fragmented packets are often used as DoS attacks. To set disallow fragments, enter the following command:

```
hostname(config)# fragment chain 1 [interface_name]
```

Enter an interface name if you want to prevent fragmentation on a specific interface. By default, this command applies to all interfaces.

Blocking Unwanted Connections

If you know that a host is attempting to attack your network (for example, system log messages show an attack), then you can block (or shun) connections based on the source IP address and other identifying parameters. No new connections can be made until you remove the shun.

Note If you have an IPS that monitors traffic, such as an AIP SSM, then the IPS can shun connections automatically.

To shun a connection manually, perform the following steps:

Step 1 If necessary, view information about the connection by entering the following command:

```
hostname# show conn
```

The security appliance shows information about each connection, such as the following:

```
TCP out 64.101.68.161:4300 in 10.86.194.60:23 idle 0:00:00 bytes 1297 flags UIO
```

Step 2 To shun connections from the source IP address, enter the following command:

```
hostname(config)# shun src_ip [dst_ip src_port dest_port [protocol]] [vlan vlan_id]
```

If you enter only the source IP address, then all future connections are shunned; existing connections remain active.

To drop an existing connection, as well as blocking future connections from the source IP address, enter the destination IP address, source and destination ports, and the protocol. By default, the protocol is 0 for IP.

For multiple context mode, you can enter this command in the admin context, and by specifying a VLAN ID that is assigned to an interface in other contexts, you can shun the connection in other contexts.

Step 3 To remove the shun, enter the following command:

```
hostname(config)# no shun src_ip [vlan vlan_id]
```

Configuring IP Audit for Basic IPS Support

The IP audit feature provides basic IPS support for a security appliance that does not have an AIP SSM. It supports a basic list of signatures, and you can configure the security appliance to perform one or more actions on traffic that matches a signature.

To enable IP audit, perform the following steps:

Step 1 To define an IP audit policy for informational signatures, enter the following command:

```
hostname(config)# ip audit name name info [action [alarm] [drop] [reset]]
```

Where **alarm** generates a system message showing that a packet matched a signature, **drop** drops the packet, and **reset** drops the packet and closes the connection. If you do not define an action, then the default action is to generate an alarm.

Step 2 To define an IP audit policy for attack signatures, enter the following command:

```
hostname(config)# ip audit name name attack [action [alarm] [drop] [reset]]
```

Where **alarm** generates a system message showing that a packet matched a signature, **drop** drops the packet, and **reset** drops the packet and closes the connection. If you do not define an action, then the default action is to generate an alarm.

Step 3 To assign the policy to an interface, enter the following command:

```
ip audit interface interface_name policy_name
```

Step 4 To disable signatures, or for more information about signatures, see the **ip audit signature** command in the *Cisco Security Appliance Command Reference*.



CHAPTER 25

Configuring QoS

Have you ever participated in a long-distance phone call that involved a satellite connection? The conversation might be interrupted with brief, but perceptible, gaps at odd intervals. Those gaps are the time, called the latency, between the arrival of packets being transmitted over the network. Some network traffic, such as voice and video, cannot tolerate long latency times. Quality of Service (QoS) is a feature that lets you give priority to critical traffic, prevent bandwidth hogging, and manage network bottlenecks to prevent packet drops.

This chapter describes how to apply QoS policies, and includes the following sections:

- QoS Overview, page 25-1
- Creating the Standard Priority Queue for an Interface, page 25-5
- Identifying Traffic for QoS Using Class Maps, page 25-8
- Creating a Policy for Standard Priority Queueing and/or Policing, page 25-9
- Creating a Policy for Traffic Shaping and Hierarchical Priority Queueing, page 25-11
- Viewing QoS Statistics, page 25-13

QoS Overview

You should consider that in an ever-changing network environment, QoS is not a one-time deployment, but an ongoing, essential part of network design.

Note QoS is only available in single context mode.

This section describes the QoS features supported by the security appliance, and includes the following topics:

- Supported QoS Features, page 25-2
- What is a Token Bucket?, page 25-2
- Policing Overview, page 25-3
- Priority Queueing Overview, page 25-3
- Traffic Shaping Overview, page 25-4
- DSCP and DiffServ Preservation, page 25-5

Supported QoS Features

The security appliance supports the following QoS features:

- Policing—To prevent individual flows from hogging the network bandwidth, you can limit the maximum bandwidth used per flow. See the "Policing Overview" section on page 25-3 for more information.
- Priority queuing—For critical traffic that cannot tolerate latency, such as Voice over IP (VoIP), you can identify traffic for Low Latency Queuing (LLQ) so that it is always transmitted ahead of other traffic. See the "Priority Queueing Overview" section on page 25-3 for more information.
- Traffic shaping—If you have a device that transmits packets at a high speed, such as a security appliance with Fast Ethernet, and it is connected to a low speed device such as a cable modem, then the cable modem is a bottleneck at which packets are frequently dropped. To manage networks with differing line speeds, you can configure the security appliance to transmit packets at a fixed slower rate. See the "Traffic Shaping Overview" section on page 25-4 for more information.

What is a Token Bucket?

A token bucket is used to manage a device that regulates the data in a flow. For example, the regulator might be a traffic policer or a traffic shaper. A token bucket itself has no discard or priority policy. Rather, a token bucket discards tokens and leaves to the flow the problem of managing its transmission queue if the flow overdrives the regulator.

A token bucket is a formal definition of a rate of transfer. It has three components: a burst size, an average rate, and a time interval. Although the average rate is generally represented as bits per second, any two values may be derived from the third by the relation shown as follows:

average rate = burst size / time interval

Here are some definitions of these terms:

- Average rate—Also called the committed information rate (CIR), it specifies how much data can be sent or forwarded per unit time on average.
- Burst size—Also called the Committed Burst (Bc) size, it specifies in bits or bytes per burst how much traffic can be sent within a given unit of time to not create scheduling concerns. (For traffic shaping, it specifies bits per burst; for policing, it specifies bytes per burst.)
- Time interval—Also called the measurement interval, it specifies the time quantum in seconds per burst.

In the token bucket metaphor, tokens are put into the bucket at a certain rate. The bucket itself has a specified capacity. If the bucket fills to capacity, newly arriving tokens are discarded. Each token is permission for the source to send a certain number of bits into the network. To send a packet, the regulator must remove from the bucket a number of tokens equal in representation to the packet size.

If not enough tokens are in the bucket to send a packet, the packet either waits until the bucket has enough tokens (in the case of traffic shaping) or the packet is discarded or marked down (in the case of policing). If the bucket is already full of tokens, incoming tokens overflow and are not available to future packets. Thus, at any time, the largest burst a source can send into the network is roughly proportional to the size of the bucket.

Note that the token bucket mechanism used for traffic shaping has both a token bucket and a data buffer, or queue; if it did not have a data buffer, it would be a policer. For traffic shaping, packets that arrive that cannot be sent immediately are delayed in the data buffer.

For traffic shaping, a token bucket permits burstiness but bounds it. It guarantees that the burstiness is bounded so that the flow will never send faster than the token bucket capacity, divided by the time interval, plus the established rate at which tokens are placed in the token bucket. See the following formula:

(token bucket capacity in bits / time interval in seconds) + established rate in bps = maximum flow speed in bps

This method of bounding burstiness also guarantees that the long-term transmission rate will not exceed the established rate at which tokens are placed in the bucket.

Policing Overview

Policing is a way of ensuring that no traffic exceeds the maximum rate (in bits/second) that you configure, thus ensuring that no one traffic flow or class can take over the entire resource. When traffic exceeds the maximum rate, the security appliance drops the excess traffic. Policing also sets the largest single burst of traffic allowed.

Priority Queueing Overview

LLQ priority queueing lets you prioritize certain traffic flows (such as latency-sensitive traffic like voice and video) ahead of other traffic.

The security appliance supports two types of priority queueing:

- Standard priority queueing—Standard priority queueing uses an LLQ priority queue on an interface (see the "Creating the Standard Priority Queue for an Interface" section on page 25-5), while all other traffic goes into the "best effort" queue. Because queues are not of infinite size, they can fill and overflow. When a queue is full, any additional packets cannot get into the queue and are dropped. This is called *tail drop*. To avoid having the queue fill up, you can increase the queue buffer size. You can also fine-tune the maximum number of packets allowed into the transmit queue. These options let you control the latency and robustness of the priority queuing. Packets in the LLQ queue are always transmitted before packets in the best effort queue.
- Hierarchical priority queueing—Hierarchical priority queueing is used on interfaces on which you enable a traffic shaping queue. A subset of the shaped traffic can be prioritized. The standard priority queue is not used. See the following guidelines about hierarchical priority queueing:
  - Priority packets are always queued at the head of the shape queue so they are always transmitted ahead of other non-priority queued packets.
  - Priority packets are never dropped from the shape queue unless the sustained rate of priority traffic exceeds the shape rate.
  - For IPSec-encrypted packets, you can only match traffic based on the DSCP or precedence setting.
  - IPSec-over-TCP is not supported for priority traffic classification.

Traffic Shaping Overview

Traffic shaping is used to match device and link speeds, thereby controlling packet loss, variable delay, and link saturation, which can cause jitter and delay.

- Traffic shaping must be applied to all outgoing traffic on a physical interface or in the case of the ASA 5505, on a VLAN. You cannot configure traffic shaping for specific types of traffic.
- Traffic shaping is implemented when packets are ready to be transmitted on an interface, so the rate calculation is performed based on the actual size of a packet to be transmitted, including all the possible overhead such as the IPSec header and L2 header.
- The shaped traffic includes both through-the-box and from-the-box traffic.
- The shape rate calculation is based on the standard token bucket algorithm. The token bucket size is twice the Burst Size value. See the "What is a Token Bucket?" section on page 25-2.
- When bursty traffic exceeds the specified shape rate, packets are queued and transmitted later. Following are some characteristics regarding the shape queue (for information about hierarchical priority queueing, see the "Priority Queueing Overview" section on page 25-3):
  - The queue size is calculated based on the shape rate. The queue can hold the equivalent of 200-milliseconds worth of shape rate traffic, assuming a 1500-byte packet. The minimum queue size is 64.
  - When the queue limit is reached, packets are tail-dropped.
  - Certain critical keep-alive packets such as OSPF Hello packets are never dropped.
  - The time interval is derived by *time_interval* = *burst_size* / *average_rate*. The larger the time interval is, the burstier the shaped traffic might be, and the longer the link might be idle. The effect can be best understood using the following exaggerated example:

    Average Rate = 1000000

    Burst Size = 1000000

    In the above example, the time interval is 1 second, which means, 1 Mbps of traffic can be bursted out within the first 10 milliseconds of the 1-second interval on a 100 Mbps FE link and leave the remaining 990 milliseconds idle without being able to send any packets until the next time interval. So if there is delay-sensitive traffic such as voice traffic, the Burst Size should be reduced compared to the average rate so the time interval is reduced.

How QoS Features Interact

You can configure each of the QoS features alone if desired for the security appliance. Often, though, you configure multiple QoS features on the security appliance so you can prioritize some traffic, for example, and prevent other traffic from causing bandwidth problems.

See the following supported feature combinations per interface:

- Standard priority queuing (for specific traffic) + Policing (for the rest of the traffic).

  You cannot configure priority queueing and policing for the same set of traffic.
- Traffic shaping (for all traffic on an interface) + Hierarchical priority queueing (for a subset of traffic).

You cannot configure traffic shaping and standard priority queueing for the same interface; only hierarchical priority queueing is allowed. For example, if you configure standard priority queueing for the global policy, and then configure traffic shaping for a specific interface, the feature you configured last is rejected because the global policy overlaps the interface policy.

Typically, if you enable traffic shaping, you do not also enable policing for the same traffic, although the security appliance does not restrict you from configuring this.

DSCP and DiffServ Preservation

- DSCP markings are preserved on all traffic passing through the security appliance.
- The security appliance does not locally mark/remark any classified traffic, but it honors the Expedited Forwarding (EF) DSCP bits of every packet to determine if it requires “priority” handling and will direct those packets to the LLQ.
- DiffServ marking is preserved on packets when they traverse the service provider backbone so that QoS can be applied in transit (QoS tunnel pre-classification).

Creating the Standard Priority Queue for an Interface

If you enable standard priority queueing for traffic on a physical interface, then you need to also create the priority queue on each interface. Each physical interface uses two queues: one for priority traffic, and the other for all other traffic. For the other traffic, you can optionally configure policing.

Note The standard priority queue is not required for hierarchical priority queueing with traffic shaping; see the “Priority Queueing Overview” section on page 25-3 for more information.

This section includes the following topics:

- Determining the Queue and TX Ring Limits, page 25-6
- Configuring the Priority Queue, page 25-7

Determining the Queue and TX Ring Limits

To determine the priority queue and TX ring limits, use the worksheets below.

Table 25-1 shows how to calculate the priority queue size. Because queues are not of infinite size, they can fill and overflow. When a queue is full, any additional packets cannot get into the queue and are dropped (called *tail drop*). To avoid having the queue fill up, you can adjust the queue buffer size according to the "Configuring the Priority Queue" section on page 25-7.

Table 25-1 Queue Limit Worksheet

Step 1	_____ Mbps	x	**125**	=	_____ # of bytes/ms			
	Outbound bandwidth (Mbps or Kbps)[1]							
	Kbps	x	**.125**	=	_____ # of bytes/ms			
Step 2	_____ *# of bytes/ms from Step 1*	÷	_____ *Average packet size (bytes)*[2]	x	_____ *Delay (ms)*[3]	=	_____ *Queue limit (# of packets)*	

1. For example, DSL might have an uplink speed of 768 Kbps.Check with your provider.
2. Determine this value from a codec or sampling size. For example, for VoIP over VPN, you might use 160 bytes. We recommend 256 bytes if you do not know what size to use.
3. The delay depends on your application. For example, the recommended maximum delay for VoIP is 200 ms. We recommend 500 ms if you do not know what delay to use.

Table 25-2 shows how to calculate the TX ring limit. This limit determines the maximum number of packets allowed into the Ethernet transmit driver before the driver pushes back to the queues on the interface to let them buffer packets until the congestion clears. This setting guarantees that the hardware-based transmit ring imposes a limited amount of extra latency for a high-priority packet.

Table 25-2 TX Ring Limit Worksheet

Step 1	_____ Mbps	x	**125**	=	_____ # of bytes/ms			
	Outbound bandwidth (Mbps or Kbps)[1]							
	Kbps	x	**0.125**	=	_____ # of bytes/ms			
Step 2	_____ *# of bytes/ms from Step 1*	÷	_____ *Maximum packet size (bytes)*[2]	x	_____ *Delay (ms)*[3]	=	_____ *TX ring limit (# of packets)*	

1. For example, DSL might have an uplink speed of 768 Kbps.Check with your provider.
2. Typically, the maximum size is 1538 bytes, or 1542 bytes for tagged Ethernet.
3. The delay depends on your application. For example, to control jitter for VoIP, you should use 20 ms.

Configuring the Priority Queue

To create the priority queue, perform the following steps.

Step 1 To create the priority queue, enter the following command:

```
hostname(config)# priority-queue interface_name
```

Where the *interface_name* argument specifies the physical interface name on which you want to enable the priority queue, or for the ASA 5505, the VLAN interface name.

Step 2 (Optional) To change the size of the priority queues, enter the following command:

```
hostname(config-priority-queue)# queue-limit number_of_packets
```

The default queue limit is 1024 packets. Because queues are not of infinite size, they can fill and overflow. When a queue is full, any additional packets cannot get into the queue and are dropped (called *tail drop*). To avoid having the queue fill up, you can use the **queue-limit** command to increase the queue buffer size.

See the "Determining the Queue and TX Ring Limits" section on page 25-6 to determine the *number_of_packets* value.

The upper limit of the range of values for the **queue-limit** command is determined dynamically at run time. To view this limit, enter **queue-limit ?** on the command line. The key determinants are the memory needed to support the queues and the memory available on the device.

The **queue-limit** that you specify affects both the higher priority low-latency queue and the best effort queue.

Step 3 (Optional) To specify the depth of the priority queues, enter the following command:

```
hostname(config-priority-queue)# tx-ring-limit number_of_packets
```

The default tx-ring-limit is 128 packets. This command sets the maximum number of low-latency or normal priority packets allowed into the Ethernet transmit driver before the driver pushes back to the queues on the interface to let them buffer packets until the congestion clears. This setting guarantees that the hardware-based transmit ring imposes a limited amount of extra latency for a high-priority packet.

See the "Determining the Queue and TX Ring Limits" section on page 25-6 to determine the *number_of_packets* value.

The upper limit of the range of values for the **tx-ring-limit** command is determined dynamically at run time. To view this limit, enter **tx-ring-limit ?** on the command line. The key determinants are the memory needed to support the queues and the memory available on the device.

The **tx-ring-limit** that you specify affects both the higher priority low-latency queue and the best-effort queue.

The following example establishes a priority queue on interface "outside" (the GigabitEthernet0/1 interface), with the default queue-limit and tx-ring-limit.

```
hostname(config)# priority-queue outside
```

The following example establishes a priority queue on the interface "outside" (the GigabitEthernet0/1 interface), sets the queue-limit to 260 packets, and sets the tx-ring-limit to 3:

```
hostname(config)# priority-queue outside
hostname(config-priority-queue)# queue-limit 260
hostname(config-priority-queue)# tx-ring-limit 3
```

Identifying Traffic for QoS Using Class Maps

QoS is part of the Modular Policy Framework. See the Chapter 16, "Using Modular Policy Framework," for more information. In Modular Policy Framework, you identify the traffic on which you want to enable QoS in a class map. This section includes the following topics:

- Creating a QoS Class Map, page 25-8
- QoS Class Map Examples, page 25-8

Creating a QoS Class Map

For priority traffic, identify only latency-sensitive traffic. For policing traffic, you can choose to police all other traffic, or you can limit the traffic to certain types. For traffic shaping, all traffic on an interface must be shaped.

To create the class maps for QoS traffic, see the **class-map** command in the "Identifying Traffic (Layer 3/4 Class Map)" section on page 16-4.

You can match traffic based on many characteristics, including access lists, tunnel groups, DSCP, precedence, and more. See the following guidelines for configuring class maps for QoS:

- For traffic shaping, you can only use the **class-default** class map, which is automatically created by the security appliance, and which matches all traffic.
- You cannot use the **class-default** class map for priority traffic.
- For hierarchical priority queueing, for IPSec-encrypted packets, you can only match traffic based on the DSCP or precedence setting.
- For hierarchical priority queueing, IPSec-over-TCP traffic is not supported.

QoS Class Map Examples

For example, in the following sequence, the **class-map** command classifies all non-tunneled TCP traffic, using an access list named tcp_traffic:

```
hostname(config)# access-list tcp_traffic permit tcp any any
hostname(config)# class-map tcp_traffic
hostname(config-cmap)# match access-list tcp_traffic
```

In the following example, other, more specific match criteria are used for classifying traffic for specific, security-related tunnel groups. These specific match criteria stipulate that a match on tunnel-group (in this case, the previously-defined Tunnel-Group-1) is required as the first match characteristic to classify traffic for a specific tunnel, and it allows for an additional match line to classify the traffic (IP differential services code point, expedited forwarding).

```
hostname(config)# class-map TG1-voice
hostname(config-cmap)# match tunnel-group tunnel-grp1
hostname(config-cmap)# match dscp ef
```

In the following example, the **class-map** command classifies both tunneled and non-tunneled traffic according to the traffic type:

```
hostname(config)# access-list tunneled extended permit ip 10.10.34.0 255.255.255.0
20.20.10.0 255.255.255.0
hostname(config)# access-list non-tunneled extended permit tcp any any
hostname(config)# tunnel-group tunnel-grp1 type IPSec_L2L
```

```
hostname(config)# class-map browse
hostname(config-cmap)# description "This class-map matches all non-tunneled tcp traffic."
hostname(config-cmap)# match access-list non-tunneled

hostname(config-cmap)# class-map TG1-voice
hostname(config-cmap)# description "This class-map matches all dscp ef traffic for
tunnel-grp 1."
hostname(config-cmap)# match dscp ef
hostname(config-cmap)# match tunnel-group tunnel-grp1

hostname(config-cmap)# class-map TG1-BestEffort
hostname(config-cmap)# description "This class-map matches all best-effort traffic for
tunnel-grp1."
hostname(config-cmap)# match tunnel-group tunnel-grp1
hostname(config-cmap)# match flow ip destination-address
```

The following example shows a way of policing a flow within a tunnel, provided the classed traffic is not specified as a tunnel, but does go *through* the tunnel. In this example, 192.168.10.10 is the address of the host machine on the private side of the remote tunnel, and the access list is named "host-over-l2l". By creating a class-map (named "host-specific"), you can then police the "host-specific" class before the LAN-to-LAN connection polices the tunnel. In this example, the "host-specific" traffic is rate-limited before the tunnel, then the tunnel is rate-limited:

```
hostname(config)# access-list host-over-l2l extended permit ip any host 192.168.10.10
hostname(config)# class-map host-specific
hostname(config-cmap)# match access-list host-over-l2l
```

The following example builds on the configuration developed in the previous section. As in the previous example, there are two named class-maps: tcp_traffic and TG1-voice.

```
hostname(config)# class-map TG1-best-effort
hostname(config-cmap)# match tunnel-group Tunnel-Group-1
hostname(config-cmap)# match flow ip destination-address
```

Adding a third class map provides a basis for defining a tunneled and non-tunneled QoS policy, as follows, which creates a simple QoS policy for tunneled and non-tunneled traffic, assigning packets of the class TG1-voice to the low latency queue and setting rate limits on the tcp_traffic and TG1-best-effort traffic flows.

Creating a Policy for Standard Priority Queueing and/or Policing

After you identify the traffic in "Identifying Traffic for QoS Using Class Maps" section on page 25-8, you can create a policy map for an interface or globally for all interfaces that assigns QoS actions (and other feature actions) to the traffic in the class map. (See the Chapter 16, "Using Modular Policy Framework," for information about other features. This chapter only discusses QoS.)

You can configure standard priority queueing and policing for different class maps within the same policy map. See the "How QoS Features Interact" section on page 25-4 for information about valid QoS configurations.

To create a policy map, perform the following steps:

Step 1 To add or edit a policy map, enter the following command:

```
hostname(config)# policy-map name
```

For example:

```
hostname(config)# policy-map QoS_policy
```

Step 2 To configure priority queueing, enter the following commands:

```
hostname(config-pmap)# class priority_map_name
hostname(config-pmap-c)# priority
```

where the *priority_map_name* is the class map you created for prioritized traffic in "Identifying Traffic for QoS Using Class Maps" section on page 25-8.

For example:

```
hostname(config)# class-map priority-class
hostname(config-cmap)# match tunnel-group Tunnel-Group-1
hostname(config-cmap)# match dscp ef

hostname(config-cmap)# policy-map QoS_policy

hostname(config-pmap)# class priority_class
hostname(config-pmap-c)# priority
```

Step 3 To configure policing, enter the following commands:

```
hostname(config-pmap)# class policing_map_name
hostname(config-pmap-c)# police {output | input} conform-rate [conform-burst]
[conform-action [drop | transmit]] [exceed-action [drop | transmit]]
```

where the *policing_map_name* is the class map you created for prioritized traffic in "Identifying Traffic for QoS Using Class Maps" section on page 25-8.

The *conform-burst argument* specifies the maximum number of instantaneous bytes allowed in a sustained burst before throttling to the conforming rate value, between 1000 and 512000000 bytes.

The **conform-action** keyword sets the action to take when the rate is less than the *conform_burst* value.

The *conform-rate* argument sets the rate limit for this traffic flow; between 8000 and 2000000000 bits per second.

The **drop** keyword drops the packet.

The **exceed-action** keyword sets the action to take when the rate is between the *conform-rate* value and the *conform-burst* value.

The **input** keyword enables policing of traffic flowing in the input direction.

The **output** keyword enables policing of traffic flowing in the output direction.

The **transmit** keyword transmits the packet.

For example:

```
hostname(config)# class-map policing-class
hostname(config-cmap)# match any

hostname(config-cmap)# policy-map QoS_policy

hostname(config-pmap)# class police_class
hostname(config-pmap-c)# police output 56000 10500
```

Step 4 To activate the policy map on one or more interfaces, enter the following command:

```
hostname(config)# service-policy policymap_name {global | interface interface_name}
```

Where **global** applies the policy map to all interfaces, and **interface** applies the policy to one interface. Only one global policy is allowed. Interface service policies take precedence over the global service policy for a given feature. For example, if you have a global policy with inspections, and an interface

policy with TCP normalization, then both inspections and TCP normalization are applied to the interface. However, if you have a global policy with inspections, and an interface policy with inspections, then only the interface policy inspections are applied to that interface.

In this example, the maximum rate for traffic of the tcp_traffic class is 56,000 bits/second and a maximum burst size of 10,500 bytes per second. For the TC1-BestEffort class, the maximum rate is 200,000 bits/second, with a maximum burst of 37,500 bytes/second. Traffic in the TC1-voice class has no policed maximum speed or burst rate because it belongs to a priority class.

```
hostname(config)# access-list tcp_traffic permit tcp any any
hostname(config)# class-map tcp_traffic
hostname(config-cmap)# match access-list tcp_traffic

hostname(config)# class-map TG1-voice
hostname(config-cmap)# match tunnel-group tunnel-grp1
hostname(config-cmap)# match dscp ef

hostname(config-cmap)# class-map TG1-BestEffort
hostname(config-cmap)# match tunnel-group tunnel-grp1
hostname(config-cmap)# match flow ip destination-address

hostname(config)# policy-map qos
hostname(config-pmap)# class tcp_traffic
hostname(config-pmap-c)# police output 56000 10500

hostname(config-pmap-c)# class TG1-voice
hostname(config-pmap-c)# priority

hostname(config-pmap-c)# class TG1-best-effort
hostname(config-pmap-c)# police output 200000 37500

hostname(config-pmap-c)# class class-default
hostname(config-pmap-c)# police output 1000000 37500

hostname(config-pmap-c)# service-policy qos global
```

Creating a Policy for Traffic Shaping and Hierarchical Priority Queueing

You can create a policy map for an interface or globally for all interfaces that assigns QoS actions (and other feature actions) to the traffic in the class map. (See the Chapter 16, “Using Modular Policy Framework,” for information about other features. This chapter only discusses QoS.)

You can configure traffic shaping for all traffic on an interface, and optionally hierarchical priority queueing for a subset of latency-sensitive traffic. See the “How QoS Features Interact” section on page 25-4 for information about valid QoS configurations.

If you want to configure hierarchical priority queueing, then first identify the traffic in “Identifying Traffic for QoS Using Class Maps” section on page 25-8; traffic shaping always uses the **class-default** class map, which is automatically available.

Note One side-effect of priority queueing is packet re-ordering. For IPSec packets, out-of-order packets that are not within the anti-replay window generate warning syslog messages. These warnings are false alarms in the case of priority queueing. You can configure the IPSec anti-replay window size to avoid possible false alarms. See the **crypto ipsec security-association replay** command in the *Cisco Security Appliance Command Reference*.

To create a policy map, perform the following steps:

Step 1 (Optional) For hierarchical priority queueing, create a policy map that applies the priority queueing action to a class map by entering the following commands:

```
hostname(config)# policy-map name
hostname(config-pmap)# class priority_map_name
hostname(config-pmap-c)# priority
```

where the *priority_map_name* is the class map you created for prioritized traffic in "Identifying Traffic for QoS Using Class Maps" section on page 25-8.

For example:

```
hostname(config)# policy-map priority-sub-policy
hostname(config-pmap)# class priority-sub-map
hostname(config-pmap-c)# priority
```

Step 2 To add or edit a policy map for traffic shaping, enter the following command:

```
hostname(config)# policy-map name
```

For example:

```
hostname(config)# policy-map shape_policy
```

Step 3 To configure traffic shaping, enter the following commands:

```
hostname(config-pmap)# class class-default
hostname(config-pmap-c)# shape average rate [burst_size]
```

where the **average** *rate* argument sets the average rate of traffic in bits per second over a given fixed time period, between 64000 and 154400000. Specify a value that is a multiple of 8000. See the "Traffic Shaping Overview" section on page 25-4 for more information about how the time period is calculated.

The *burst_size* argument sets the average burst size in bits that can be transmitted over a given fixed time period, between 2048 and 154400000. Specify a value that is a multiple of 128. If you do not specify the *burst_size*, the default value is equivalent to 4-milliseconds of traffic at the specified average rate. For example, if the average rate is 1000000 bits per second, 4 ms worth = 1000000 * 4/1000 = 4000.

You can only identify the **class-default** class map, which is defined as **match any**, because the security appliance requires all traffic to be matched for traffic shaping.

Step 4 (Optional) To configure hierarchical priority queueing, enter the following command:

```
hostname(config-pmap-c)# service-policy priority_policy_map_name
```

where the *priority_policy_map_name* is the policy map you created for prioritized traffic in Step 1.

For example:

```
hostname(config)# policy-map priority-sub-policy
hostname(config-pmap)# class priority-sub-map
hostname(config-pmap-c)# priority
```

```
hostname(config-pmap-c)# policy-map shape_policy
hostname(config-pmap)# class class-default
hostname(config-pmap-c)# shape
hostname(config-pmap-c)# service-policy priority-sub-policy
```

Step 5 To activate the policy map on an interface, enter the following command:

```
hostname(config)# service-policy policymap_name interface interface_name
```

Note You cannot configure traffic shaping in the global policy.

The following example enables traffic shaping on the outside interface, and limits traffic to 2 Mbps; priority queueing is enabled for VoIP traffic that is tagged with DSCP EF and AF13 and for IKE traffic:

```
hostname(config)# access-list ike permit udp any any eq 500
hostname(config)# class-map ike
hostname(config-cmap)# match access-list ike

hostname(config-cmap)# class-map voice_traffic
hostname(config-cmap)# match dscp EF AF13

hostname(config-cmap)# policy-map qos_class_policy
hostname(config-pmap)# class voice_traffic
hostname(config-pmap-c)# priority
hostname(config-pmap-c)# class ike
hostname(config-pmap-c)# priority

hostname(config-pmap-c)# policy-map qos_outside_policy
hostname(config-pmap)# class class-default
hostname(config-pmap-c)# shape average 2000000 16000
hostname(config-pmap-c)# service-policy qos_class_policy

hostname(config-pmap-c)# service-policy qos_outside_policy interface outside
```

Viewing QoS Statistics

This section includes the following topics:

- Viewing QoS Police Statistics, page 25-13
- Viewing QoS Standard Priority Statistics, page 25-14
- Viewing QoS Shaping Statistics, page 25-14
- Viewing QoS Standard Priority Queue Statistics, page 25-15

Viewing QoS Police Statistics

To view the QoS statistics for traffic policing, use the **show service-policy** command with the **police** keyword:

```
hostname# show service-policy police
```

The following is sample output for the **show service-policy police** command:

```
hostname# show service-policy police

Global policy:
    Service-policy: global_fw_policy

Interface outside:
    Service-policy: qos
        Class-map: browse
            police Interface outside:
                cir 56000 bps, bc 10500 bytes
                conformed 10065 packets, 12621510 bytes; actions: transmit
                exceeded 499 packets, 625146 bytes; actions: drop
                conformed 5600 bps, exceed 5016 bps
        Class-map: cmap2
            police Interface outside:
                cir 200000 bps, bc 37500 bytes
                conformed 17179 packets, 20614800 bytes; actions: transmit
                exceeded 617 packets, 770718 bytes; actions: drop
                conformed 198785 bps, exceed 2303 bps
```

Viewing QoS Standard Priority Statistics

To view statistics for service policies implementing the **priority** command, use the **show service-policy** command with the **priority** keyword:

```
hostname# show service-policy priority
```

The following is sample output for the **show service-policy priority** command:

```
hostname# show service-policy priority
Global policy:
    Service-policy: global_fw_policy
Interface outside:
    Service-policy: qos
        Class-map: TG1-voice
            Priority:
                Interface outside: aggregate drop 0, aggregate transmit 9383
```

Note "Aggregate drop" denotes the aggregated drop in this interface; "aggregate transmit" denotes the aggregated number of transmitted packets in this interface.

Viewing QoS Shaping Statistics

To view statistics for service policies implementing the **shape** command, use the **show service-policy** command with the **shape** keyword:

```
hostname# show service-policy shape
```

The following is sample output for the **show service-policy shape** command:

```
hostname# show service-policy shape
Interface outside
  Service-policy: shape
    Class-map: class-default

      Queueing
      queue limit 64 packets
```

```
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0

      shape (average) cir 2000000, bc 8000, be 8000
```

The following is sample output of the **show service policy shape** command, which includes service policies that include the **shape** command and the **service-policy** command that calls the hierarchical priority policy and the related statistics:

```
hostname# show service-policy shape

Interface outside:
  Service-policy: shape
    Class-map: class-default

      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0

      shape (average) cir 2000000, bc 16000, be 16000

      Service-policy: voip
        Class-map: voip

          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0
        Class-map: class-default

          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0
```

Viewing QoS Standard Priority Queue Statistics

To display the priority-queue statistics for an interface, use the **show priority-queue statistics** command in privileged EXEC mode. The results show the statistics for both the best-effort (BE) queue and the low-latency queue (LLQ). The following example shows the use of the **show priority-queue statistics** command for the interface named test, and the command output.

```
hostname# show priority-queue statistics test

Priority-Queue Statistics interface test

Queue Type         = BE
Packets Dropped    = 0
Packets Transmit   = 0
Packets Enqueued   = 0
Current Q Length   = 0
Max Q Length       = 0

Queue Type         = LLQ
Packets Dropped    = 0
Packets Transmit   = 0
Packets Enqueued   = 0
Current Q Length   = 0
Max Q Length       = 0
hostname#
```

In this statistical report, the meaning of the line items is as follows:

- “Packets Dropped” denotes the overall number of packets that have been dropped in this queue.
- “Packets Transmit” denotes the overall number of packets that have been transmitted in this queue.
- “Packets Enqueued” denotes the overall number of packets that have been queued in this queue.
- “Current Q Length” denotes the current depth of this queue.
- “Max Q Length” denotes the maximum depth that ever occurred in this queue.



CHAPTER 26

Configuring Application Layer Protocol Inspection

This chapter describes how to configure application layer protocol inspection. Inspection engines are required for services that embed IP addressing information in the user data packet or that open secondary channels on dynamically assigned ports. These protocols require the security appliance to do a deep packet inspection instead of passing the packet through the fast path (see the "Stateful Inspection Overview" section on page 1-17 for more information about the fast path). As a result, inspection engines can affect overall throughput.

Several common inspection engines are enabled on the security appliance by default, but you might need to enable others depending on your network. This chapter includes the following sections:

- Inspection Engine Overview, page 26-2
  - When to Use Application Protocol Inspection, page 26-2
  - Inspection Limitations, page 26-2
  - Default Inspection Policy, page 26-3
- Configuring Application Inspection, page 26-5
- CTIQBE Inspection, page 26-10
- DCERPC Inspection, page 26-12
- DNS Inspection, page 26-14
- ESMTP Inspection, page 26-24
- FTP Inspection, page 26-27
- GTP Inspection, page 26-33
- H.323 Inspection, page 26-39
- HTTP Inspection, page 26-46
- Instant Messaging Inspection, page 26-50
- ICMP Inspection, page 26-53
- ICMP Error Inspection, page 26-54
- ILS Inspection, page 26-54
- MGCP Inspection, page 26-55
- MMP Inspection, page 26-59
- NetBIOS Inspection, page 26-61

- PPTP Inspection, page 26-63
- RADIUS Accounting Inspection, page 26-63
- RSH Inspection, page 26-64
- RTSP Inspection, page 26-65
- SIP Inspection, page 26-68
- Skinny (SCCP) Inspection, page 26-75
- SMTP and Extended SMTP Inspection, page 26-78
- SNMP Inspection, page 26-80
- SQL*Net Inspection, page 26-80
- Sun RPC Inspection, page 26-81
- TFTP Inspection, page 26-83
- XDMCP Inspection, page 26-84

Inspection Engine Overview

This section includes the following topics:

- When to Use Application Protocol Inspection, page 26-2
- Inspection Limitations, page 26-2
- Default Inspection Policy, page 26-3

When to Use Application Protocol Inspection

When a user establishes a connection, the security appliance checks the packet against access lists, creates an address translation, and creates an entry for the session in the fast path, so that further packets can bypass time-consuming checks. However, the fast path relies on predictable port numbers and does not perform address translations inside a packet.

Many protocols open secondary TCP or UDP ports. The initial session on a well-known port is used to negotiate dynamically assigned port numbers.

Other applications embed an IP address in the packet that needs to match the source address that is normally translated when it goes through the security appliance.

If you use applications like these, then you need to enable application inspection.

When you enable application inspection for a service that embeds IP addresses, the security appliance translates embedded addresses and updates any checksum or other fields that are affected by the translation.

When you enable application inspection for a service that uses dynamically assigned ports, the security appliance monitors sessions to identify the dynamic port assignments, and permits data exchange on these ports for the duration of the specific session.

Inspection Limitations

See the following limitations for application protocol inspection:

- State information for multimedia sessions that require inspection are not passed over the state link for stateful failover. The exception is GTP, which is replicated over the state link.
- Some inspection engines do not support PAT, NAT, outside NAT, or NAT between same security interfaces. See "Default Inspection Policy" for more information about NAT support.
- For all the application inspections, the adaptive security appliance limits the number of simultaneous, active data connections to 200 connections. For example, if an FTP client opens multiple secondary connections, the FTP inspection engine allows only 200 active connections and the 201 connection is dropped and the adaptive security appliance generates a system error message.

Default Inspection Policy

By default, the configuration includes a policy that matches all default application inspection traffic and applies inspection to the traffic on all interfaces (a global policy). Default application inspection traffic includes traffic to the default ports for each protocol. You can only apply one global policy, so if you want to alter the global policy, for example, to apply inspection to non-standard ports, or to add inspections that are not enabled by default, you need to either edit the default policy or disable it and apply a new one.

Table 26-1 lists all inspections supported, the default ports used in the default class map, and the inspection engines that are on by default, shown in bold. This table also notes any NAT limitations.

Table 26-1 Supported Application Inspection Engines

Application[1]	Default Port	NAT Limitations	Standards[2]	Comments
CTIQBE	TCP/2748	—	—	—
DNS over UDP	UDP/53	No NAT support is available for name resolution through WINS.	RFC 1123	No PTR records are changed.
FTP	TCP/21	—	RFC 959	—
GTP	UDP/3386 UDP/2123	—	—	Requires a special license.
H.323 H.225 and **RAS**	TCP/1720 UDP/1718 UDP (RAS) 1718-1719	No NAT on same security interfaces. No static PAT.	ITU-T H.323, H.245, H225.0, Q.931, Q.932	—
HTTP	TCP/80	—	RFC 2616	Beware of MTU limitations stripping ActiveX and Java. If the MTU is too small to allow the Java or ActiveX tag to be included in one packet, stripping may not occur.
ICMP	—	—	—	All ICMP traffic is matched in the default class map.
ICMP ERROR	—	—	—	All ICMP traffic is matched in the default class map.
ILS (LDAP)	TCP/389	No PAT.	—	—
MGCP	UDP/2427, 2727	—	RFC 2705bis-05	—

Table 26-1 Supported Application Inspection Engines (continued)

Application[1]	Default Port	NAT Limitations	Standards[2]	Comments
MMP	TCP/TLS 5443	There are no embedded NAT or secondary connections.	—	—
NetBIOS Name Server over IP	UDP/137, 138 (Source ports)	—	—	NetBIOS is supported by performing NAT of the packets for NBNS UDP port 137 and NBDS UDP port 138.
PPTP	TCP/1723	—	RFC 2637	—
RADIUS Accounting	1646	—	RFC 2865	—
RSH	TCP/514	No PAT	Berkeley UNIX	—
RTSP	TCP/554	No PAT. No outside NAT.	RFC 2326, 2327, 1889	No handling for HTTP cloaking.
SIP	TCP/5060 UDP/5060	No outside NAT. No NAT on same security interfaces.	RFC 3261	—
SKINNY (SCCP)	TCP/2000	No outside NAT. No NAT on same security interfaces.	—	Does not handle TFTP uploaded Cisco IP Phone configurations under certain circumstances.
SMTP and **ESMTP**	TCP/25	—	RFC 821, 1123	—
SNMP	UDP/161, 162	No NAT or PAT.	RFC 1155, 1157, 1212, 1213, 1215	v.2 RFC 1902-1908; v.3 RFC 2570-2580.
SQL*Net	TCP/1521	—	—	v.1 and v.2.
Sun RPC over UDP and TCP	UDP/111	No NAT or PAT.	—	The default class map includes UDP port 111; if you want to enable Sun RPC inspection for TCP port 111, you need to create a new class map that matches TCP port 111, add the class to the policy, and then apply the **inspect sunrpc** command to that class.
TFTP	UDP/69	—	RFC 1350	Payload IP addresses are not translated.
XDCMP	UDP/177	No NAT or PAT.	—	—

1. Inspection engines that are enabled by default for the default port are in bold.
2. The security appliance is in compliance with these standards, but it does not enforce compliance on packets being inspected. For example, FTP commands are supposed to be in a particular order, but the security appliance does not enforce the order.

The default policy configuration includes the following commands:

```
class-map inspection_default
 match default-inspection-traffic
policy-map type inspect dns preset_dns_map
 parameters
  message-length maximum 512
policy-map global_policy
 class inspection_default
  inspect dns preset_dns_map
```

```
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect rsh
  inspect rtsp
  inspect esmtp
  inspect sqlnet
  inspect skinny
  inspect sunrpc
  inspect xdmcp
  inspect sip
  inspect netbios
  inspect tftp
service-policy global_policy global
```

Configuring Application Inspection

This feature uses Modular Policy Framework, so that implementing application inspection consists of identifying traffic, applying inspections to the traffic, and activating inspections on an interface. For some applications, you can perform special actions when you enable inspection. See Chapter 16, "Using Modular Policy Framework," for more information.

Inspection is enabled by default for some applications. See the "Default Inspection Policy" section for more information. Use this section to modify your inspection policy.

To configure application inspection, perform the following steps:

Step 1 To identify the traffic to which you want to apply inspections, add either a Layer 3/4 class map for through traffic or a Layer 3/4 class map for management traffic. See the "Creating a Layer 3/4 Class Map for Through Traffic" section on page 16-5 and "Creating a Layer 3/4 Class Map for Management Traffic" section on page 16-7 for detailed information. The management Layer 3/4 class map can be used only with the RADIUS accounting inspection.

The default Layer 3/4 class map for through traffic is called "inspection_default." It matches traffic using a special **match** command, **match default-inspection-traffic**, to match the default ports for each application protocol.

You can specify a **match access-list** command along with the **match default-inspection-traffic** command to narrow the matched traffic to specific IP addresses. Because the **match default-inspection-traffic** command specifies the ports to match, any ports in the access list are ignored.

If you want to match non-standard ports, then create a new class map for the non-standard ports. See the "Default Inspection Policy" section on page 26-3 for the standard ports for each inspection engine. You can combine multiple class maps in the same policy if desired, so you can create one class map to match certain traffic, and another to match different traffic. However, if traffic matches a class map that contains an inspection command, and then matches another class map that also has an inspection command, only the first matching class is used. For example, SNMP matches the inspection_default class. To enable SNMP inspection, enable SNMP inspection for the default class in Step 5. Do not add another class that matches SNMP.

For example, to limit inspection to traffic from 10.1.1.0 to 192.168.1.0 using the default class map, enter the following commands:

```
hostname(config)# access-list inspect extended permit ip 10.1.1.0 255.255.255.0
192.168.1.0 255.255.255.0
hostname(config)# class-map inspection_default
hostname(config-cmap)# match access-list inspect
```

View the entire class map using the following command:

```
hostname(config-cmap)# show running-config class-map inspection_default
!
class-map inspection_default
 match default-inspection-traffic
 match access-list inspect
!
```

To inspect FTP traffic on port 21 as well as 1056 (a non-standard port), create an access list that specifies the ports, and assign it to a new class map:

```
hostname(config)# access-list ftp_inspect extended permit tcp any any eq 21
hostname(config)# access-list ftp_inspect extended permit tcp any any eq 1056
hostname(config)# class-map new_inspection
hostname(config-cmap)# match access-list ftp_inspect
```

Step 2 (Optional) Some inspection engines let you control additional parameters when you apply the inspection to the traffic. See the following sections to configure an inspection policy map for your application:

- DCERPC—See the "Configuring a DCERPC Inspection Policy Map for Additional Inspection Control" section on page 26-13
- DNS—See the "Configuring a DNS Inspection Policy Map for Additional Inspection Control" section on page 26-21
- ESMTP—See the "Configuring an ESMTP Inspection Policy Map for Additional Inspection Control" section on page 26-25
- FTP—See the "Configuring an FTP Inspection Policy Map for Additional Inspection Control" section on page 26-29.
- GTP—See the "Configuring a GTP Inspection Policy Map for Additional Inspection Control" section on page 26-34.
- H323—See the "Configuring an H.323 Inspection Policy Map for Additional Inspection Control" section on page 26-41
- HTTP—See the "Configuring an HTTP Inspection Policy Map for Additional Inspection Control" section on page 26-46.
- Instant Messaging—See the "Configuring an Instant Messaging Inspection Policy Map for Additional Inspection Control" section on page 26-50
- MGCP—See the "Configuring an MGCP Inspection Policy Map for Additional Inspection Control" section on page 26-57.
- NetBIOS—See the "Configuring a NetBIOS Inspection Policy Map for Additional Inspection Control" section on page 26-61
- RADIUS Accounting—See the "Configuring a RADIUS Inspection Policy Map for Additional Inspection Control" section on page 26-64
- RTSP—See the "Configuring an RTSP Inspection Policy Map for Additional Inspection Control" section on page 26-66
- SIP—See the "Configuring a SIP Inspection Policy Map for Additional Inspection Control" section on page 26-70
- Skinny—See the "Configuring a Skinny (SCCP) Inspection Policy Map for Additional Inspection Control" section on page 26-77
- SNMP—See the "SNMP Inspection" section on page 26-80.

Step 3 To add or edit a Layer 3/4 policy map that sets the actions to take with the class map traffic, enter the following command:

```
hostname(config)# policy-map name
hostname(config-pmap)#
```

The default policy map is called "global_policy." This policy map includes the default inspections listed in the "Default Inspection Policy" section on page 26-3. If you want to modify the default policy (for example, to add or delete an inspection, or to identify an additional class map for your actions), then enter **global_policy** as the name.

Step 4 To identify the class map from Step 1 to which you want to assign an action, enter the following command:

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

If you are editing the default policy map, it includes the inspection_default class map. You can edit the actions for this class by entering **inspection_default** as the name. To add an additional class map to this policy map, identify a different name. You can combine multiple class maps in the same policy if desired, so you can create one class map to match certain traffic, and another to match different traffic. However, if traffic matches a class map that contains an inspection command, and then matches another class map that also has an inspection command, only the first matching class is used. For example, SNMP matches the inspection_default class map.To enable SNMP inspection, enable SNMP inspection for the default class in Step 5. Do not add another class that matches SNMP.

Step 5 Enable application inspection by entering the following command:

```
hostname(config-pmap-c)# inspect protocol
```

The *protocol* is one of the following values:

Table 26-2 Protocol Keywords

Keywords	Notes
ctiqbe	—
dcerpc [*map_name*]	If you added a DCERPC inspection policy map according to "Configuring a DCERPC Inspection Policy Map for Additional Inspection Control" section on page 26-13, identify the map name in this command.
dns [*map_name*]	If you added a DNS inspection policy map according to "Configuring a DNS Inspection Policy Map for Additional Inspection Control" section on page 26-21, identify the map name in this command. The default DNS inspection policy map name is "preset_dns_map." The default inspection policy map sets the maximum DNS packet length to 512 bytes.
esmtp [*map_name*]	If you added an ESMTP inspection policy map according to "Configuring an ESMTP Inspection Policy Map for Additional Inspection Control" section on page 26-25, identify the map name in this command.

Table 26-2 Protocol Keywords

Keywords	Notes
ftp [**strict** [*map_name*]]	Use the **strict** keyword to increase the security of protected networks by preventing web browsers from sending embedded commands in FTP requests. See the "Using the strict Option" section on page 26-28 for more information. If you added an FTP inspection policy map according to "Configuring an FTP Inspection Policy Map for Additional Inspection Control" section on page 26-29, identify the map name in this command.
gtp [*map_name*]	If you added a GTP inspection policy map according to the "Configuring a GTP Inspection Policy Map for Additional Inspection Control" section on page 26-34, identify the map name in this command.
h323 h225 [*map_name*]	If you added an H323 inspection policy map according to "Configuring an H.323 Inspection Policy Map for Additional Inspection Control" section on page 26-41, identify the map name in this command.
h323 ras [*map_name*]	If you added an H323 inspection policy map according to "Configuring an H.323 Inspection Policy Map for Additional Inspection Control" section on page 26-41, identify the map name in this command.
http [*map_name*]	If you added an HTTP inspection policy map according to the "Configuring an HTTP Inspection Policy Map for Additional Inspection Control" section on page 26-46, identify the map name in this command.
icmp	—
icmp error	—
ils	—
im [*map_name*]	If you added an Instant Messaging inspection policy map according to "Configuring an Instant Messaging Inspection Policy Map for Additional Inspection Control" section on page 26-50, identify the map name in this command.
mgcp [*map_name*]	If you added an MGCP inspection policy map according to "Configuring an MGCP Inspection Policy Map for Additional Inspection Control" section on page 26-57, identify the map name in this command.
mmp tls-proxy [*name*]	—
netbios [*map_name*]	If you added a NetBIOS inspection policy map according to "Configuring a NetBIOS Inspection Policy Map for Additional Inspection Control" section on page 26-61, identify the map name in this command.
pptp	—

Table 26-2 Protocol Keywords

Keywords	Notes
radius-accounting [*map_name*]	The **radius-accounting** keyword is only available for a management class map. See the "Creating a Layer 3/4 Class Map for Management Traffic" section on page 16-7 for more information about creating a management class map. If you added a RADIUS accounting inspection policy map according to "Configuring a RADIUS Inspection Policy Map for Additional Inspection Control" section on page 26-64, identify the map name in this command.
rsh	—
rtsp [*map_name*]	If you added a NetBIOS inspection policy map according to "Configuring an RTSP Inspection Policy Map for Additional Inspection Control" section on page 26-66, identify the map name in this command.
sip [*map_name*]	If you added a SIP inspection policy map according to "Configuring a SIP Inspection Policy Map for Additional Inspection Control" section on page 26-70, identify the map name in this command.
skinny [*map_name*]	If you added a Skinny inspection policy map according to "Configuring a Skinny (SCCP) Inspection Policy Map for Additional Inspection Control" section on page 26-77, identify the map name in this command.
snmp [*map_name*]	If you added an SNMP inspection policy map according to "SNMP Inspection" section on page 26-80, identify the map name in this command.
sqlnet	—
sunrpc	The default class map includes UDP port 111; if you want to enable Sun RPC inspection for TCP port 111, you need to create a new class map that matches TCP port 111, add the class to the policy, and then apply the **inspect sunrpc** command to that class.
tftp	—
xdmcp	—

Step 6 To activate the policy map on one or more interfaces, enter the following command:

```
hostname(config)# service-policy policymap_name {global | interface interface_name}
```

Where **global** applies the policy map to all interfaces, and **interface** applies the policy to one interface. By default, the default policy map, "global_policy," is applied globally. Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.

CTIQBE Inspection

This section describes CTIQBE application inspection. This section includes the following topics:

- CTIQBE Inspection Overview, page 26-10
- Limitations and Restrictions, page 26-10
- Verifying and Monitoring CTIQBE Inspection, page 26-10

CTIQBE Inspection Overview

CTIQBE protocol inspection supports NAT, PAT, and bidirectional NAT. This enables Cisco IP SoftPhone and other Cisco TAPI/JTAPI applications to work successfully with Cisco CallManager for call setup across the security appliance.

TAPI and JTAPI are used by many Cisco VoIP applications. CTIQBE is used by Cisco TSP to communicate with Cisco CallManager.

Limitations and Restrictions

The following summarizes limitations that apply when using CTIQBE application inspection:

- CTIQBE application inspection does not support configurations with the **alias** command.
- Stateful failover of CTIQBE calls is not supported.
- Entering the **debug ctiqbe** command may delay message transmission, which may have a performance impact in a real-time environment. When you enable this debugging or logging and Cisco IP SoftPhone seems unable to complete call setup through the security appliance, increase the timeout values in the Cisco TSP settings on the system running Cisco IP SoftPhone.

The following summarizes special considerations when using CTIQBE application inspection in specific scenarios:

- If two Cisco IP SoftPhones are registered with different Cisco CallManagers, which are connected to different interfaces of the security appliance, calls between these two phones fails.
- When Cisco CallManager is located on the higher security interface compared to Cisco IP SoftPhones, if NAT or outside NAT is required for the Cisco CallManager IP address, the mapping must be static as Cisco IP SoftPhone requires the Cisco CallManager IP address to be specified explicitly in its Cisco TSP configuration on the PC.
- When using PAT or Outside PAT, if the Cisco CallManager IP address is to be translated, its TCP port 2748 must be statically mapped to the same port of the PAT (interface) address for Cisco IP SoftPhone registrations to succeed. The CTIQBE listening port (TCP 2748) is fixed and is not user-configurable on Cisco CallManager, Cisco IP SoftPhone, or Cisco TSP.

Verifying and Monitoring CTIQBE Inspection

The **show ctiqbe** command displays information regarding the CTIQBE sessions established across the security appliance. It shows information about the media connections allocated by the CTIQBE inspection engine.

The following is sample output from the **show ctiqbe** command under the following conditions. There is only one active CTIQBE session setup across the security appliance. It is established between an internal CTI device (for example, a Cisco IP SoftPhone) at local address 10.0.0.99 and an external Cisco CallManager at 172.29.1.77, where TCP port 2748 is the Cisco CallManager. The heartbeat interval for the session is 120 seconds.

```
hostname# # show ctiqbe

Total: 1
        LOCAL           FOREIGN         STATE   HEARTBEAT
-----------------------------------------------------------------
1       10.0.0.99/1117  172.29.1.77/2748        1       120
        ---------------------------------------------
        RTP/RTCP: PAT xlates: mapped to 172.29.1.99(1028 - 1029)
        ---------------------------------------------
        MEDIA: Device ID 27     Call ID 0
               Foreign 172.29.1.99      (1028 - 1029)
               Local   172.29.1.88      (26822 - 26823)
        ---------------------------------------------
```

The CTI device has already registered with the CallManager. The device internal address and RTP listening port is PATed to 172.29.1.99 UDP port 1028. Its RTCP listening port is PATed to UDP 1029.

The line beginning with `RTP/RTCP: PAT xlates:` appears only if an internal CTI device has registered with an external CallManager and the CTI device address and ports are PATed to that external interface. This line does not appear if the CallManager is located on an internal interface, or if the internal CTI device address and ports are translated to the same external interface that is used by the CallManager.

The output indicates a call has been established between this CTI device and another phone at 172.29.1.88. The RTP and RTCP listening ports of the other phone are UDP 26822 and 26823. The other phone locates on the same interface as the CallManager because the security appliance does not maintain a CTIQBE session record associated with the second phone and CallManager. The active call leg on the CTI device side can be identified with Device ID 27 and Call ID 0.

The following is sample output from the **show xlate debug** command for these CTIBQE connections:

```
hostname# show xlate debug
3 in use, 3 most used
Flags:  D - DNS, d - dump, I - identity, i - inside, n - no random,
        r - portmap, s - static
TCP PAT from inside:10.0.0.99/1117 to outside:172.29.1.99/1025 flags ri idle 0:00:22
timeout 0:00:30
UDP PAT from inside:10.0.0.99/16908 to outside:172.29.1.99/1028 flags ri idle 0:00:00
timeout 0:04:10
UDP PAT from inside:10.0.0.99/16909 to outside:172.29.1.99/1029 flags ri idle 0:00:23
timeout 0:04:10
```

The **show conn state ctiqbe** command displays the status of CTIQBE connections. In the output, the media connections allocated by the CTIQBE inspection engine are denoted by a ‘C’ flag. The following is sample output from the **show conn state ctiqbe** command:

```
hostname# show conn state ctiqbe
1 in use, 10 most used
hostname# show conn state ctiqbe detail
1 in use, 10 most used
Flags: A - awaiting inside ACK to SYN, a - awaiting outside ACK to SYN,
       B - initial SYN from outside, C - CTIQBE media, D - DNS, d - dump,
       E - outside back connection, F - outside FIN, f - inside FIN,
       G - group, g - MGCP, H - H.323, h - H.225.0, I - inbound data,
       i - incomplete, J - GTP, j - GTP data, k - Skinny media,
       M - SMTP data, m - SIP media, O - outbound data, P - inside back connection,
       q - SQL*Net data, R - outside acknowledged FIN,
       R - UDP RPC, r - inside acknowledged FIN, S - awaiting inside SYN,
```

```
s - awaiting outside SYN, T - SIP, t - SIP transient, U - up
```

DCERPC Inspection

This section describes the DCERPC inspection engine. This section includes the following topics:

- DCERPC Overview, page 26-12
- Configuring a DCERPC Inspection Policy Map for Additional Inspection Control, page 26-13

DCERPC Overview

DCERPC is a protocol widely used by Microsoft distributed client and server applications that allows software clients to execute programs on a server remotely.

DCERPC inspection maps inspection for native TCP communication between a server called the Endpoint Mapper (EPM) and client on the well-known TCP port 135. Map and lookup operations of the EPM are supported for clients. Client and server can be located in any security zone. The embedded server IP address and port number are received from the applicable EPM response messages. Because a client can attempt multiple connections to the server port returned by EPM, creation of multiple pinholes is allowed. User configurable timeouts are allowed for multiple pinholes.

Note DCERPC inspection only supports communication between an EPM server and clients to open pinholes through the security appliance. Clients using RPC communication that does not use an EPM server is not supported with DCERPC inspection.

Typically, software clients remotely execute programs on an EPM server in the following way:

Step 1 A client queries an EPM server for the dynamically-allocated port number of a required DCERPC service. The EPM server listens on the well-known TCP port 135.

Step 2 The security appliance, located between the client and EPM server, intercepts the communication.

Step 3 The EPM server indicates the port number on which the DCERPC service is available.

Step 4 The security appliance opens a pinhole for that DCERPC service.

Note Because the pinhole does not have a value for the source port, the source port value is set to 0. The source IP address, destination IP address, and destination port are indicated.

Step 5 Using that pinhole, the client attempts to connect to the DCERPC service on the indicated port.

Step 6 The security appliance detects that the connection is permitted and creates a secondary connection to the server instance providing the DCERPC service. When creating the secondary connection, the security appliance applies NAT if necessary.

Note When the pinhole timeout is reached, the security appliance destroys the pinhole and permits new connections from the client to the EPM server; existing connections remain, however, because they are independent.

Configuring a DCERPC Inspection Policy Map for Additional Inspection Control

To specify additional DCERPC inspection parameters, create a DCERPC inspection policy map. You can then apply the inspection policy map when you enable DCERPC inspection according to the "Configuring Application Inspection" section on page 26-5.

To create a DCERPC inspection policy map, perform the following steps:

Step 1 Create a DCERPC inspection policy map, enter the following command:

```
hostname(config)# policy-map type inspect dcerpc policy_map_name
hostname(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 2 (Optional) To add a description to the policy map, enter the following command:

```
hostname(config-pmap)# description string
```

Step 3 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

b. To configure the timeout for DCERPC pinholes and override the global system pinhole timeout of two minutes, enter the following command:

```
hostname(config-pmap-p)# timeout pinhole hh:mm:ss
```

Where the *hh:mm:ss* argument is the timeout for pinhole connections. Value is between 0:0:1 and 1193:0:0.

c. To configure options for the endpoint mapper traffic, enter the following command:

```
hostname(config-pmap-p)# endpoint-mapper [epm-service-only] [lookup-operation
[timeout hh:mm:ss]]
```

Where the *hh:mm:ss* argument is the timeout for pinholes generated from the lookup operation. If no timeout is configured for the lookup operation, the timeout pinhole command or the default is used. The **epm-service-only** keyword enforces endpoint mapper service during binding so that only its service traffic is processed. The **lookup-operation** keyword enables the lookup operation of the endpoint mapper service.

The following example shows how to define a DCERPC inspection policy map with the timeout configured for DCERPC pinholes.

```
hostname(config)# policy-map type inspect dcerpc dcerpc_map
```

```
hostname(config-pmap)# timeout pinhole 0:10:00

hostname(config)# class-map dcerpc
hostname(config-cmap)# match port tcp eq 135

hostname(config)# policy-map global-policy
hostname(config-pmap)# class dcerpc
hostname(config-pmap-c)# inspect dcerpc dcerpc-map

hostname(config)# service-policy global-policy global
```

DNS Inspection

This section describes DNS application inspection. This section includes the following topics:

- How DNS Application Inspection Works, page 26-14
- How DNS Rewrite Works, page 26-15
- Configuring DNS Rewrite, page 26-16
- Verifying and Monitoring DNS Inspection, page 26-21

How DNS Application Inspection Works

The security appliance tears down the DNS session associated with a DNS query as soon as the DNS reply is forwarded by the security appliance. The security appliance also monitors the message exchange to ensure that the ID of the DNS reply matches the ID of the DNS query.

When DNS inspection is enabled, which is the default, the security appliance performs the following additional tasks:

- Translates the DNS record based on the configuration completed using the **alias**, **static** and **nat** commands (DNS Rewrite). Translation only applies to the A-record in the DNS reply; therefore, DNS Rewrite does not affect reverse lookups, which request the PTR record.

Note DNS Rewrite is not applicable for PAT because multiple PAT rules are applicable for each A-record and the PAT rule to use is ambiguous.

- Enforces the maximum DNS message length (the default is 512 bytes and the maximum length is 65535 bytes). The security appliance performs reassembly as needed to verify that the packet length is less than the maximum length configured. The security appliance drops the packet if it exceeds the maximum length.

Note If you enter the **inspect dns** command without the **maximum-length** option, DNS packet size is not checked

- Enforces a domain-name length of 255 bytes and a label length of 63 bytes.
- Verifies the integrity of the domain-name referred to by the pointer if compression pointers are encountered in the DNS message.
- Checks to see if a compression pointer loop exists.

A single connection is created for multiple DNS sessions, as long as they are between the same two hosts, and the sessions have the same 5-tuple (source/destination IP address, source/destination port, and protocol). DNS identification is tracked by *app_id*, and the idle timer for each app_id runs independently.

Because the app_id expires independently, a legitimate DNS response can only pass through the security appliance within a limited period of time and there is no resource build-up. However, if you enter the **show conn** command, you will see the idle timer of a DNS connection being reset by a new DNS session. This is due to the nature of the shared DNS connection and is by design.

How DNS Rewrite Works

When DNS inspection is enabled, DNS rewrite provides full support for NAT of DNS messages originating from any interface.

If a client on an inside network requests DNS resolution of an inside address from a DNS server on an outside interface, the DNS A-record is translated correctly. If the DNS inspection engine is disabled, the A-record is not translated.

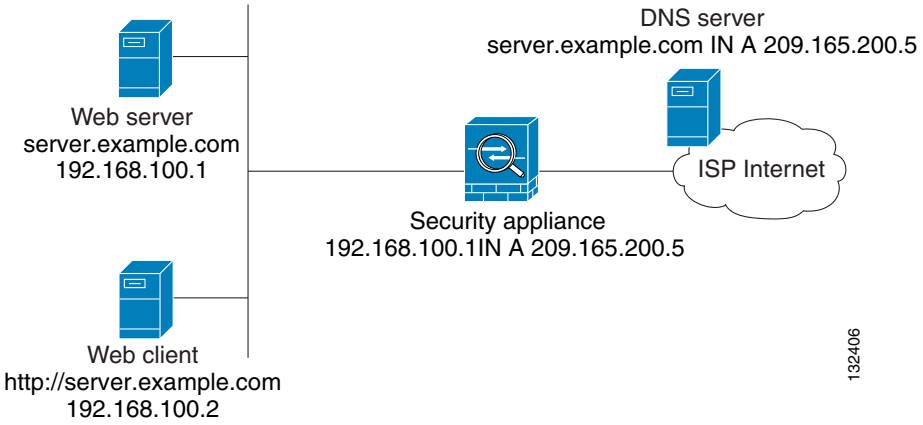
As long as DNS inspection remains enabled, you can configure DNS rewrite using the **alias**, **static**, or **nat** commands. For details about the configuration required see the "Configuring DNS Rewrite" section on page 26-16.

DNS Rewrite performs two functions:

- Translating a public address (the routable or "mapped" address) in a DNS reply to a private address (the "real" address) when the DNS client is on a private interface.
- Translating a private address to a public address when the DNS client is on the public interface.

In Figure 26-1, the DNS server resides on the external (ISP) network The real address of the server (192.168.100.1) has been mapped using the **static** command to the ISP-assigned address (209.165.200.5). When a web client on the inside interface attempts to access the web server with the URL http://server.example.com, the host running the web client sends a DNS request to the DNS server to resolve the IP address of the web server. The security appliance translates the non-routable source address in the IP header and forwards the request to the ISP network on its outside interface. When the DNS reply is returned, the security appliance applies address translation not only to the destination address, but also to the embedded IP address of the web server, which is contained in the A-record in the DNS reply. As a result, the web client on the inside network gets the correct address for connecting to the web server on the inside network. For configuration instructions for scenarios similar to this one, see the "Configuring DNS Rewrite with Two NAT Zones" section on page 26-17.

Figure 26-1 Translating the Address in a DNS Reply (DNS Rewrite)



DNS rewrite also works if the client making the DNS request is on a DMZ network and the DNS server is on an inside interface. For an illustration and configuration instructions for this scenario, see the "DNS Rewrite with Three NAT Zones" section on page 26-18.

Configuring DNS Rewrite

You configure DNS rewrite using the **alias**, **static**, or **nat** commands. The **alias** and **static** command can be used interchangeably; however, we recommend using the **static** command for new deployments because it is more precise and unambiguous. Also, DNS rewrite is optional when using the **static** command.

This section describes how to use the **alias** and **static** commands to configure DNS rewrite. It provides configuration procedures for using the **static** command in a simple scenario and in a more complex scenario. Using the **nat** command is similar to using the **static** command except that DNS Rewrite is based on dynamic translation instead of a static mapping.

This section includes the following topics:

- Using the Static Command for DNS Rewrite, page 26-16
- Using the Static Command for DNS Rewrite, page 26-16
- Configuring DNS Rewrite with Two NAT Zones, page 26-17
- DNS Rewrite with Three NAT Zones, page 26-18
- Configuring DNS Rewrite with Three NAT Zones, page 26-20

For detailed syntax and additional functions for the **alias**, **nat**, and **static** command, see the appropriate command page in the *Cisco Security Appliance Command Reference*.

Using the Static Command for DNS Rewrite

The **static** command causes addresses on an IP network residing on a specific interface to be translated into addresses on another IP network on a different interface. The syntax for this command is as follows:

```
hostname(config)# static (real_ifc,mapped_ifc) mapped-address real-address dns
```

The following example specifies that the address 192.168.100.10 on the inside interface is translated into 209.165.200.5 on the outside interface:

```
hostname(config)# static (inside,outside) 209.165.200.225 192.168.100.10 dns
```

Note Using the **nat** command is similar to using the **static** command except that DNS Rewrite is based on dynamic translation instead of a static mapping.

Using the Alias Command for DNS Rewrite

The **alias** command causes the security appliance to translate addresses on an IP network residing on any interface into addresses on another IP network connected through a different interface. The syntax for this command is as follows:

```
hostname(config)# alias (interface_name) mapped-address real-address
```

The following example specifies that the real address (192.168.100.10) on any interface except the inside interface will be translated to the mapped address (**209.165.200.225**) on the inside interface. Notice that the location of 192.168.100.10 is not precisely defined.

```
hostname(config)# alias (inside) 209.165.200.225 192.168.100.10
```

Note If you use the **alias** command to configure DNS Rewrite, proxy ARP will be performed for the mapped address. To prevent this, disable Proxy ARP by entering the **sysopt noproxyarp** command after entering the **alias** command.

Configuring DNS Rewrite with Two NAT Zones

To implement a DNS Rewrite scenario similar to the one shown in Figure 26-1, perform the following steps:

Step 1 Create a static translation for the web server, as follows:

```
hostname(config)# static (real_ifc,mapped_ifc) mapped-address real-address netmask
255.255.255.255 dns
```

where the arguments are as follows:

- *real_ifc*—The name of the interface connected to the real addresses.
- *mapped_ifc*—The name of the interface where you want the addresses to be mapped.
- *mapped-address*—The translated IP address of the web server.
- *real-address*—The real IP address of the web server.

Step 2 Create an access list that permits traffic to the port that the web server listens to for HTTP requests.

```
hostname(config)# access-list acl-name extended permit tcp any host mapped-address eq port
```

where the arguments are as follows:

acl-name—The name you give the access list.

mapped-address—The translated IP address of the web server.

port—The TCP port that the web server listens to for HTTP requests.

Step 3 Apply the access list created in Step 2 to the mapped interface. To do so, use the **access-group** command, as follows:

```
hostname(config)# access-group acl-name in interface mapped_ifc
```

Step 4 If DNS inspection is disabled or if you want to change the maximum DNS packet length, configure DNS inspection. DNS application inspection is enabled by default with a maximum DNS packet length of 512 bytes. For configuration instructions, see the "Configuring Application Inspection" section on page 26-5.

Step 5 On the public DNS server, add an A-record for the web server, such as:

```
domain-qualified-hostname. IN A mapped-address
```

where *domain-qualified-hostname* is the hostname with a domain suffix, as in server.example.com. The period after the hostname is important. *mapped-address* is the translated IP address of the web server.

The following example configures the security appliance for the scenario shown in Figure 26-1. It assumes DNS inspection is already enabled.

```
hostname(config)# static (inside,outside) 209.165.200.225 192.168.100.1 netmask
255.255.255.255 dns
hostname(config)# access-list 101 permit tcp any host 209.165.200.225 eq www
hostname(config)# access-group 101 in interface outside
```

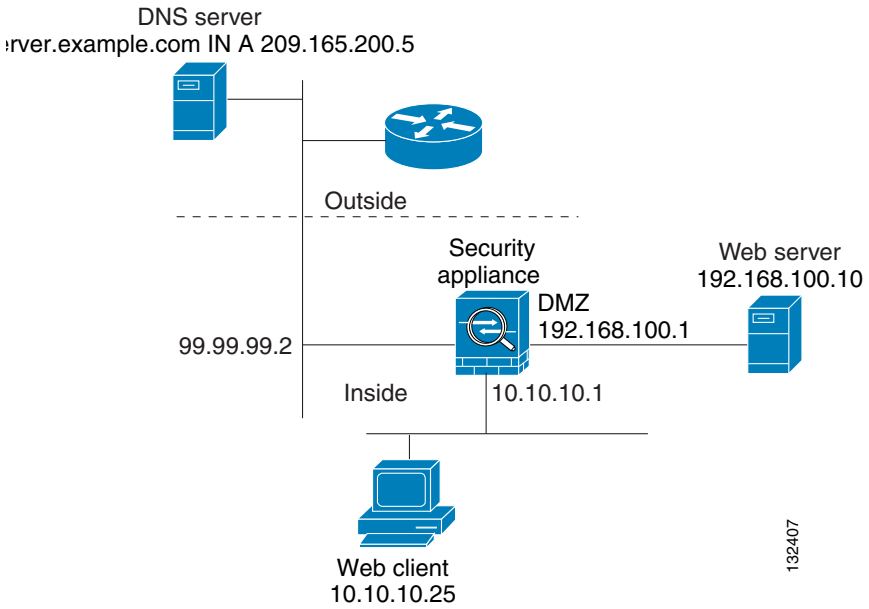
This configuration requires the following A-record on the DNS server:

```
server.example.com. IN A 209.165.200.225
```

DNS Rewrite with Three NAT Zones

Figure 26-2 provides a more complex scenario to illustrate how DNS inspection allows NAT to operate transparently with a DNS server with minimal configuration. For configuration instructions for scenarios like this one, see the "Configuring DNS Rewrite with Three NAT Zones" section on page 26-20.

Figure 26-2 DNS Rewrite with Three NAT Zones



In Figure 26-2, a web server, server.example.com, has the real address 192.168.100.10 on the DMZ interface of the security appliance. A web client with the IP address 10.10.10.25 is on the inside interface and a public DNS server is on the outside interface. The site NAT policies are as follows:

- The outside DNS server holds the authoritative address record for server.example.com.
- Hosts on the outside network can contact the web server with the domain name server.example.com through the outside DNS server or with the IP address 209.165.200.5.
- Clients on the inside network can access the web server with the domain name server.example.com through the outside DNS server or with the IP address 192.168.100.10.

When a host or client on any interface accesses the DMZ web server, it queries the public DNS server for the A-record of server.example.com. The DNS server returns the A-record showing that server.example.com binds to address 209.165.200.5.

When a web client on the *outside* network attempts to access http://server.example.com, the sequence of events is as follows:

1. The host running the web client sends the DNS server a request for the IP address of server.example.com.
2. The DNS server responds with the IP address 209.165.200.225 in the reply.
3. The web client sends its HTTP request to 209.165.200.225.
4. The packet from the outside host reaches the security appliance at the outside interface.
5. The static rule translates the address 209.165.200.225 to 192.168.100.10 and the security appliance directs the packet to the web server on the DMZ.

When a web client on the *inside* network attempts to access http://server.example.com, the sequence of events is as follows:

1. The host running the web client sends the DNS server a request for the IP address of server.example.com.
2. The DNS server responds with the IP address 209.165.200.225 in the reply.
3. The security appliance receives the DNS reply and submits it to the DNS application inspection engine.
4. The DNS application inspection engine does the following:
   a. Searches for any NAT rule to undo the translation of the embedded A-record address “[outside]:209.165.200.5”. In this example, it finds the following static configuration:

      ```
      static (dmz,outside) 209.165.200.225 192.168.100.10 dns
      ```

   b. Uses the static rule to rewrite the A-record as follows because the **dns** option is included:

      ```
      [outside]:209.165.200.225 --> [dmz]:192.168.100.10
      ```

      Note If the **dns** option were not included with the **static** command, DNS Rewrite would not be performed and other processing for the packet continues.

   c. Searches for any NAT to translate the web server address, [dmz]:192.168.100.10, when communicating with the inside web client.

      No NAT rule is applicable, so application inspection completes.

      If a NAT rule (nat or static) were applicable, the **dns** option must also be specified. If the **dns** option were not specified, the A-record rewrite in step b would be reverted and other processing for the packet continues.

5. The security appliance sends the HTTP request to server.example.com on the DMZ interface.

Configuring DNS Rewrite with Three NAT Zones

To enable the NAT policies for the scenario in Figure 26-2, perform the following steps:

Step 1 Create a static translation for the web server on the DMZ network, as follows:

```
hostname(config)# static (dmz,outside) mapped-address real-address dns
```

where the arguments are as follows:

- *dmz*—The name of the DMZ interface of the security appliance.
- *outside*—The name of the outside interface of the security appliance.
- *mapped-address*—The translated IP address of the web server.
- *real-address*—The real IP address of the web server.

Step 2 Create an access list that permits traffic to the port that the web server listens to for HTTP requests.

```
hostname(config)# access-list acl-name extended permit tcp any host mapped-address eq port
```

where the arguments are as follows:

acl-name—The name you give the access list.

mapped-address—The translated IP address of the web server.

port—The TCP port that the web server listens to for HTTP requests.

Step 3 Apply the access list created in Step 2 to the outside interface. To do so, use the **access-group** command, as follows:

```
hostname(config)# access-group acl-name in interface outside
```

Step 4 If DNS inspection is disabled or if you want to change the maximum DNS packet length, configure DNS inspection. DNS application inspection is enabled by default with a maximum DNS packet length of 512 bytes. For configuration instructions, see the "Configuring Application Inspection" section on page 26-5.

Step 5 On the public DNS server, add an A-record for the web server, such as:

```
domain-qualified-hostname. IN A mapped-address
```

where *domain-qualified-hostname* is the hostname with a domain suffix, as in server.example.com. The period after the hostname is important. *mapped-address* is the translated IP address of the web server.

The following example configures the security appliance for the scenario shown in Figure 26-2. It assumes DNS inspection is already enabled.

```
hostname(config)# static (dmz,outside) 209.165.200.225 192.168.100.10 dns
hostname(config)# access-list 101 permit tcp any host 209.165.200.225 eq www
hostname(config)# access-group 101 in interface outside
```

This configuration requires the following A-record on the DNS server:

```
server.example.com. IN A 209.165.200.225
```

Verifying and Monitoring DNS Inspection

To view information about the current DNS connections, enter the following command:

```
hostname# show conn
```

For connections using a DNS server, the source port of the connection may be replaced by the IP address of DNS server in the show conn command output.

A single connection is created for multiple DNS sessions, as long as they are between the same two hosts, and the sessions have the same 5-tuple (source/destination IP address, source/destination port, and protocol). DNS identification is tracked by app_id, and the idle timer for each app_id runs independently.

Because the app_id expires independently, a legitimate DNS response can only pass through the security appliance within a limited period of time and there is no resource build-up. However, when you enter the **show conn** command, you see the idle timer of a DNS connection being reset by a new DNS session. This is due to the nature of the shared DNS connection and is by design.

To display the statistics for DNS application inspection, enter the **show service-policy** command. The following is sample output from the **show service-policy** command:

```
hostname# show service-policy
Interface outside:
  Service-policy: sample_policy
    Class-map: dns_port
      Inspect: dns maximum-length 1500, packet 0, drop 0, reset-drop 0
```

Configuring a DNS Inspection Policy Map for Additional Inspection Control

DNS application inspection supports DNS message controls that provide protection against DNS spoofing and cache poisoning. User configurable rules allow filtering based on DNS header, domain name, resource record type and class. Zone transfer can be restricted between servers with this function, for example.

The Recursion Desired and Recursion Available flags in the DNS header can be masked to protect a public server from attack if that server only supports a particular internal zone. In addition, DNS randomization can be enabled avoid spoofing and cache poisoning of servers that either do not support randomization, or utilize a weak pseudo random number generator. Limiting the domain names that can be queried also restricts the domain names which can be queried, which protects the public server further.

A configurable DNS mismatch alert can be used as notification if an excessive number of mismatching DNS responses are received, which could indicate a cache poisoning attack. In addition, a configurable check to enforce a Transaction Signature be attached to all DNS messages is also supported.

To specify actions when a message violates a parameter, create a DNS inspection policy map. You can then apply the inspection policy map when you enable DNS inspection according to the "Configuring Application Inspection" section on page 26-5.

To create a DNS inspection policy map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the "Creating a Regular Expression" section on page 16-13. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the "Creating a Regular Expression Class Map" section on page 16-16.

Step 3 (Optional) Create a DNS inspection class map by performing the following steps.

A class map groups multiple traffic matches. Traffic must match *all* of the **match** commands to match the class map. You can alternatively identify **match** commands directly in the policy map. The difference between creating a class map and defining the traffic match directly in the inspection policy map is that the class map lets you create more complex match criteria, and you can reuse class maps.

To specify traffic that should not match the class map, use the **match not** command. For example, if the **match not** command specifies the string "example.com," then any traffic that includes "example.com" does not match the class map.

For the traffic that you identify in this class map, you can specify actions such as drop, drop-connection, reset, mask, set the rate limit, and/or log the connection in the inspection policy map.

If you want to perform different actions for each **match** command, you should identify the traffic directly in the policy map.

a. Create the class map by entering the following command:

```
hostname(config)# class-map type inspect dns [match-all | match-any] class_map_name
hostname(config-cmap)#
```

Where *class_map_name* is the name of the class map. The **match-all** keyword is the default, and specifies that traffic must match all criteria to match the class map. The **match-any** keyword specifies that the traffic matches the class map if it matches at least one of the criteria. The CLI enters class-map configuration mode, where you can enter one or more **match** commands.

b. (Optional) To add a description to the class map, enter the following command:

```
hostname(config-cmap)# description string
```

c. (Optional) To match a specific flag that is set in the DNS header, enter the following command:

```
hostname(config-cmap)# match [not] header-flag [eq] {f_well_known | f_value}
```

Where the *f_well_known* argument is the DNS flag bit. The *f_value* argument is the 16-bit value in hex. The **eq** keyword specifies an exact match.

d. (Optional) To match a DNS type, including Query type and RR type, enter the following command:

```
hostname(config-cmap)# match [not] dns-type {eq t_well_known | t_val} {range t_val1
t_val2}
```

Where the *t_well_known* argument is the DNS flag bit. The *t_val* arguments are arbitrary values in the DNS type field (0-65535). The **range** keyword specifies a range and the **eq** keyword specifies an exact match.

e. (Optional) To match a DNS class, enter the following command:

```
hostname(config-cmap)# match [not] dns-class {eq c_well_known | c_val} {range c_val1
c_val2}
```

Where the *c_well_known* argument is the DNS class. The *c_val* arguments are arbitrary values in the DNS class field. The **range** keyword specifies a range and the **eq** keyword specifies an exact match.

f. (Optional) To match a DNS question or resource record, enter the following command:

```
hostname(config-cmap)# match {question | {resource-record answer | authority | any}}
```

Where the **question** keyword specifies the question portion of a DNS message. The **resource-record** keyword specifies the resource record portion of a DNS message. The ***answer*** keyword specifies the Answer RR section. The **authority** keyword specifies the Authority RR section. The ***additional*** keyword specifies the Additional RR section.

g. (Optional) To match a DNS message domain name list, enter the following command:

```
hostname(config-cmap)# match [not] domain-name {regex regex_id | regex class class_id]
```

The **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

Step 4 Create a DNS inspection policy map, enter the following command:

```
hostname(config)# policy-map type inspect dns policy_map_name
hostname(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 5 (Optional) To add a description to the policy map, enter the following command:

```
hostname(config-pmap)# description string
```

Step 6 To apply actions to matching traffic, perform the following steps.

a. Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the DNS class map that you created in Step 3 by entering the following command:

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

- Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
hostname(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

Not all options are available for each **match** or **class** command. See the CLI help or the *Cisco Security Appliance Command Reference* for the exact options available.

The **drop** keyword drops all packets that match.

The **send-protocol-error** keyword sends a protocol error message.

The **drop-connection** keyword drops the packet and closes the connection.

The **mask** keyword masks out the matching portion of the packet.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

The **rate-limit** *message_rate* argument limits the rate of messages.

You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 16-9.

Step 7 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

b. To randomize the DNS identifier for a DNS query, enter the following command:

```
hostname(config-pmap-p)# id-randomization
```

c. To enable logging for excessive DNS ID mismatches, enter the following command:

```
hostname(config-pmap-p)# id-mismatch [count number duration seconds] action log
```

Where the **count** *string* argument specifies the maximum number of mismatch instances before a system message log is sent. The **duration** ***seconds*** specifies the period, in seconds, to monitor.

d. To require a TSIG resource record to be present, enter the following command:

```
hostname(config-pmap-p)# tsig enforced action {drop [log] | [log}
```

Where the **count** *string* argument specifies the maximum number of mismatch instances before a system message log is sent. The **duration** ***seconds*** specifies the period, in seconds, to monitor.

The following example shows a how to define a DNS inspection policy map.

```
hostname(config)# regex domain_example "example\.com"
hostname(config)# regex domain_foo "foo\.com"

hostname(config)# ! define the domain names that the server serves
hostname(config)# class-map type inspect regex match-any my_domains
hostname(config-cmap)# match regex domain_example
hostname(config-cmap)# match regex domain_foo

hostname(config)# ! Define a DNS map for query only
hostname(config)# class-map type inspect dns match-all pub_server_map
hostname(config-cmap)# match not header-flag QR
hostname(config-cmap)# match question
hostname(config-cmap)# match not domain-name regex class my_domains

hostname(config)# policy-map type inspect dns serv_prot
hostname(config-pmap)# class pub_server_map
hostname(config-pmap-c)# drop log
hostname(config-pmap-c)# match header-flag RD
hostname(config-pmap-c)# mask log

hostname(config)# class-map dns_serv_map
hostname(config-cmap)# match default-inspection-traffic

hostname(config)# policy-map pub_policy
hostname(config-pmap)# class dns_serv_map
hostname(config-pmap-c)# inspect dns serv_prot

hostname(config)# service-policy pub_policy interface dmz
```

ESMTP Inspection

ESMTP inspection detects attacks, including spam, phising, malformed message attacks, buffer overflow/underflow attacks. It also provides support for application security and protocol conformance, which enforce the sanity of the ESMTP messages as well as detect several attacks, block senders/receivers, and block mail relay.

Configuring an ESMTP Inspection Policy Map for Additional Inspection Control

To specify actions when a message violates a parameter, create an ESMTP inspection policy map. You can then apply the inspection policy map when you enable ESMTP inspection according to the "Configuring Application Inspection" section on page 26-5.

To create an ESMTP inspection policy map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the "Creating a Regular Expression" section on page 16-13. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the "Creating a Regular Expression Class Map" section on page 16-16.

Step 3 Create an ESMTP inspection policy map, enter the following command:

```
hostname(config)# policy-map type inspect esmtp policy_map_name
hostname(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 4 (Optional) To add a description to the policy map, enter the following command:

```
hostname(config-pmap)# description string
```

Step 5 To apply actions to matching traffic, perform the following steps.

a. Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
hostname(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

Not all options are available for each **match** or **class** command. See the CLI help or the *Cisco Security Appliance Command Reference* for the exact options available.

The **drop** keyword drops all packets that match.

The **send-protocol-error** keyword sends a protocol error message.

The **drop-connection** keyword drops the packet and closes the connection.

The **mask** keyword masks out the matching portion of the packet.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

The **rate-limit** *message_rate* argument limits the rate of messages.

You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 16-9.

Step 6 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

b. To configure a local domain name, enter the following command:

```
hostname(config-pmap-p)# mail-relay domain-name action [drop-connection | log]]
```

Where the **drop-connection** action closes the connection. The **log** action sends a system log message when this policy map matches traffic.

c. To enforce banner obfuscation, enter the following command:

```
hostname(config-pmap-p)# mask-banner
```

d. (Optional) To detect special characters in sender or receiver email addresses, enter the following command:

```
hostname(config-pmap-p)# special-character action [drop-connection | log]]
```

Using this command detects pipe (|), backquote (`) and null characters.

e. (Optional) To match the body length or body line length, enter the following command:

```
hostname(config-pmap-p)# match body [line] length gt length
```

Where *length* is the length of the message body or the length of a line in the message body.

f. (Optional) To match an ESMTP command verb, enter the following command:

```
hostname(config-pmap-p)# match cmd verb verb
```

Where *verb* is any of the following ESMTP commands:

```
AUTH|DATA|EHLO|ETRN||HELO|HELP|MAIL|NOOP|QUIT|RCPT|RSET|SAML|SOML|VRFY
```

g. (Optional) To match the number of recipient addresses, enter the following command:

```
hostname(config-pmap-p)# match cmd RCPT count gt count
```

Where *count* is the number of recipient addresses.

h. (Optional) To match the command line length, enter the following command:

```
hostname(config-pmap-p)# match cmd line length gt length
```

Where *length* is the command line length.

i. (Optional) To match the ehlo-reply-parameters, enter the following command:

```
hostname(config-pmap-p)# match ehlo-reply-parameter extensions
```

Where *extensions* are the ESMTP service extensions sent by the server in response to the EHLO message from the client. These extensions are implemented as a new command or as parameters to an existing command. *extensions* can be any of the following:

```
8bitmime|binarymime|checkpoint|dsn|ecode|etrn|others|pipelining|size|vrfy
```

j. (Optional) To match the header length or header line length, enter the following command:

```
hostname(config-pmap-p)# match header [line] length gt length
```

Where *length* is the number of characters in the header or line.

k. (Optional) To match the header to-fields count, enter the following command:

```
hostname(config-pmap-p)# match header to-fields count gt count
```

Where *count* is the number of recipients in the to-field of the header.

l. (Optional) To match the number of invalid recipients, enter the following command:

```
hostname(config-pmap-p)# match invalid-recipients count gt count
```

Where *count* is the number of invalid recipients.

m. (Optional) To match the type of MIME encoding scheme used, enter the following command:

```
hostname(config-pmap-p)# match mime encoding [7bit|8bit|base64|binary|others|
quoted-printable]
```

n. (Optional) To match the MIME filename length, enter the following command:

```
hostname(config-pmap-p)# match mime filename length gt length
```

Where *length* is the length of the *filename* in the range 1 to 1000.

o. (Optional) To match the MIME file type, enter the following command:

```
hostname(config-pmap-p)# match mime filetype regex [name | class name]
```

Where *name* or *class name* is the regular expression that matches a file type or a class map. The regular expression used to match a class map can select multiple file types.

p. (Optional) To match a sender address, enter the following command:

```
hostname(config-pmap-p)# match sender-address regex [name | class name]
```

Where *name* or *class name* is the regular expression that matches a sender address or a class map. The regular expression used to match a class map can select multiple sender addresses.

q. (Optional) To match the length of a sender's address, enter the following command:

```
hostname(config-pmap-p)# match sender-address length gt length
```

Where *length* is the number of characters in the sender's address.

The following example shows how to define an ESMTP inspection policy map.

```
hostname(config)# regex user1 "user1@cisco.com"
hostname(config)# regex user2 "user2@cisco.com"
hostname(config)# regex user3 "user3@cisco.com"
hostname(config)# class-map type regex senders_black_list
hostname(config-cmap)# description "Regular expressions to filter out undesired senders"
hostname(config-cmap)# match regex user1
hostname(config-cmap)# match regex user2
hostname(config-cmap)# match regex user3

hostname(config)# policy-map type inspect esmtp advanced_esmtp_map
hostname(config-pmap)# match sender-address regex class senders_black_list
hostname(config-pmap-c)# drop-connection log

hostname(config)# policy-map outside_policy
hostname(config-pmap)# class inspection_default
hostname(config-pmap-c)# inspect esmtp advanced_esmtp_map

hostname(config)# service-policy outside_policy interface outside
```

FTP Inspection

This section describes the FTP inspection engine. This section includes the following topics:

- FTP Inspection Overview, page 26-28
- Using the strict Option, page 26-28
- Configuring an FTP Inspection Policy Map for Additional Inspection Control, page 26-29
- Verifying and Monitoring FTP Inspection, page 26-32

FTP Inspection Overview

The FTP application inspection inspects the FTP sessions and performs four tasks:

- Prepares dynamic secondary data connection
- Tracks the FTP command-response sequence
- Generates an audit trail
- Translates the embedded IP address

FTP application inspection prepares secondary channels for FTP data transfer. Ports for these channels are negotiated through PORT or PASV commands. The channels are allocated in response to a file upload, a file download, or a directory listing event.

Note If you disable FTP inspection engines with the **no inspect ftp** command, outbound users can start connections only in passive mode, and all inbound FTP is disabled.

Using the strict Option

Using the **strict** option with the **inspect ftp** command increases the security of protected networks by preventing web browsers from sending embedded commands in FTP requests.

Note To specify FTP commands that are not permitted to pass through the security appliance, create an FTP map according to the "Configuring an FTP Inspection Policy Map for Additional Inspection Control" section on page 26-29.

After you enable the **strict** option on an interface, FTP inspection enforces the following behavior:

- An FTP command must be acknowledged before the security appliance allows a new command.
- The security appliance drops connections that send embedded commands.
- The 227 and PORT commands are checked to ensure they do not appear in an error string.

Caution Using the **strict** option may cause the failure of FTP clients that are not strictly compliant with FTP RFCs.

If the **strict** option is enabled, each FTP command and response sequence is tracked for the following anomalous activity:

- Truncated command—Number of commas in the PORT and PASV reply command is checked to see if it is five. If it is not five, then the PORT command is assumed to be truncated and the TCP connection is closed.

- Incorrect command—Checks the FTP command to see if it ends with <CR><LF> characters, as required by the RFC. If it does not, the connection is closed.
- Size of RETR and STOR commands—These are checked against a fixed constant. If the size is greater, then an error message is logged and the connection is closed.
- Command spoofing—The PORT command should always be sent from the client. The TCP connection is denied if a PORT command is sent from the server.
- Reply spoofing—PASV reply command (227) should always be sent from the server. The TCP connection is denied if a PASV reply command is sent from the client. This prevents the security hole when the user executes "227 xxxxx a1, a2, a3, a4, p1, p2."
- TCP stream editing—The security appliance closes the connection if it detects TCP stream editing.
- Invalid port negotiation—The negotiated dynamic port value is checked to see if it is less than 1024. As port numbers in the range from 1 to 1024 are reserved for well-known connections, if the negotiated port falls in this range, then the TCP connection is freed.
- Command pipelining—The number of characters present after the port numbers in the PORT and PASV reply command is cross checked with a constant value of 8. If it is more than 8, then the TCP connection is closed.
- The security appliance replaces the FTP server response to the SYST command with a series of Xs. to prevent the server from revealing its system type to FTP clients. To override this default behavior, use the **no mask-syst-reply** command in the FTP map.

Configuring an FTP Inspection Policy Map for Additional Inspection Control

FTP command filtering and security checks are provided using strict FTP inspection for improved security and control. Protocol conformance includes packet length checks, delimiters and packet format checks, command terminator checks, and command validation.

Blocking FTP based on user values is also supported so that it is possible for FTP sites to post files for download, but restrict access to certain users. You can block FTP connections based on file type, server name, and other attributes. System message logs are generated if an FTP connection is denied after inspection.

If you want FTP inspection to allow FTP servers to reveal their system type to FTP clients, and limit the allowed FTP commands, then create and configure an FTP map. You can then apply the FTP map when you enable FTP inspection according to the "Configuring Application Inspection" section on page 26-5.

To create an FTP map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the "Creating a Regular Expression" section on page 16-13. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the "Creating a Regular Expression Class Map" section on page 16-16.

Step 3 (Optional) Create an FTP inspection class map by performing the following steps.

A class map groups multiple traffic matches. Traffic must match *all* of the **match** commands to match the class map. You can alternatively identify **match** commands directly in the policy map. The difference between creating a class map and defining the traffic match directly in the inspection policy map is that the class map lets you create more complex match criteria, and you can reuse class maps.

To specify traffic that should not match the class map, use the **match not** command. For example, if the **match not** command specifies the string "example.com," then any traffic that includes "example.com" does not match the class map.

For the traffic that you identify in this class map, you can specify actions such as drop, drop-connection, reset, mask, set the rate limit, and/or log the connection in the inspection policy map.

If you want to perform different actions for each **match** command, you should identify the traffic directly in the policy map.

a. Create the class map by entering the following command:

```
hostname(config)# class-map type inspect ftp [match-all | match-any] class_map_name
hostname(config-cmap)#
```

Where *class_map_name* is the name of the class map. The **match-all** keyword is the default, and specifies that traffic must match all criteria to match the class map. The **match-any** keyword specifies that the traffic matches the class map if it matches at least one of the criteria. The CLI enters class-map configuration mode, where you can enter one or more **match** commands.

b. (Optional) To add a description to the class map, enter the following command:

```
hostname(config-cmap)# description string
```

c. (Optional) To match a filename for FTP transfer, enter the following command:

```
hostname(config-cmap)# match [not] filename regex [regex_name |
class regex_class_name]
```

Where the *regex_name* is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

d. (Optional) To match a file type for FTP transfer, enter the following command:

```
hostname(config-cmap)# match [not] filetype regex [regex_name |
class regex_class_name]
```

Where the *regex_name* is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

e. (Optional) To disallow specific FTP commands, use the following command:

```
hostname(config-cmap)# match [not] request-command ftp_command [ftp_command...]
```

Where *ftp_command* with one or more FTP commands that you want to restrict. See Table 26-3 for a list of the FTP commands that you can restrict.

Table 26-3 FTP Map request-command deny Options

request-command deny Option	Purpose
appe	Disallows the command that appends to a file.
cdup	Disallows the command that changes to the parent directory of the current working directory.
dele	Disallows the command that deletes a file on the server.
get	Disallows the client command for retrieving a file from the server.
help	Disallows the command that provides help information.
mkd	Disallows the command that makes a directory on the server.
put	Disallows the client command for sending a file to the server.

Table 26-3 FTP Map request-command deny Options (continued)

request-command deny Option	Purpose
rmd	Disallows the command that deletes a directory on the server.
rnfr	Disallows the command that specifies rename-from filename.
rnto	Disallows the command that specifies rename-to filename.
site	Disallows the command that are specific to the server system. Usually used for remote administration.
stou	Disallows the command that stores a file using a unique file name.

f. (Optional) To match an FTP server, enter the following command:

```
hostname(config-cmap)# match [not] server regex [regex_name | class regex_class_name]
```

Where the *regex_name* is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

g. (Optional) To match an FTP username, enter the following command:

```
hostname(config-cmap)# match [not] username regex [regex_name |
class regex_class_name]
```

Where the *regex_name* is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

Step 4 Create an FTP inspection policy map, enter the following command:

```
hostname(config)# policy-map type inspect ftp policy_map_name
hostname(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 5 (Optional) To add a description to the policy map, enter the following command:

```
hostname(config-pmap)# description string
```

Step 6 To apply actions to matching traffic, perform the following steps.

a. Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the FTP class map that you created in Step 3 by entering the following command:

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

- Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
hostname(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

Not all options are available for each **match** or **class** command. See the CLI help or the *Cisco Security Appliance Command Reference* for the exact options available.

The **drop** keyword drops all packets that match.

The **send-protocol-error** keyword sends a protocol error message.

The **drop-connection** keyword drops the packet and closes the connection.

The **mask** keyword masks out the matching portion of the packet.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

The **rate-limit** *message_rate* argument limits the rate of messages.

You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 16-9.

Step 7 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

b. To mask the greeting banner from the FTP server, enter the following command:

```
hostname(config-pmap-p)# mask-banner
```

c. To mask the reply to **syst** command, enter the following command:

```
hostname(config-pmap-p)# mask-syst-reply
```

Before submitting a username and password, all FTP users are presented with a greeting banner. By default, this banner includes version information useful to hackers trying to identify weaknesses in a system. The following example shows how to mask this banner:

```
hostname(config)# policy-map type inspect ftp mymap
hostname(config-pmap)# parameters
hostname(config-pmap-p)# mask-banner

hostname(config)# class-map match-all ftp-traffic
hostname(config-cmap)# match port tcp eq ftp

hostname(config)# policy-map ftp-policy
hostname(config-pmap)# class ftp-traffic
hostname(config-pmap-c)# inspect ftp strict mymap

hostname(config)# service-policy ftp-policy interface inside
```

Verifying and Monitoring FTP Inspection

FTP application inspection generates the following log messages:

- An Audit record 302002 is generated for each file that is retrieved or uploaded.
- The FTP command is checked to see if it is RETR or STOR and the retrieve and store commands are logged.
- The username is obtained by looking up a table providing the IP address.

- The username, source IP address, destination IP address, NAT address, and the file operation are logged.
- Audit record 201005 is generated if the secondary dynamic channel preparation failed due to memory shortage.

In conjunction with NAT, the FTP application inspection translates the IP address within the application payload. This is described in detail in RFC 959.

GTP Inspection

This section describes the GTP inspection engine. This section includes the following topics:

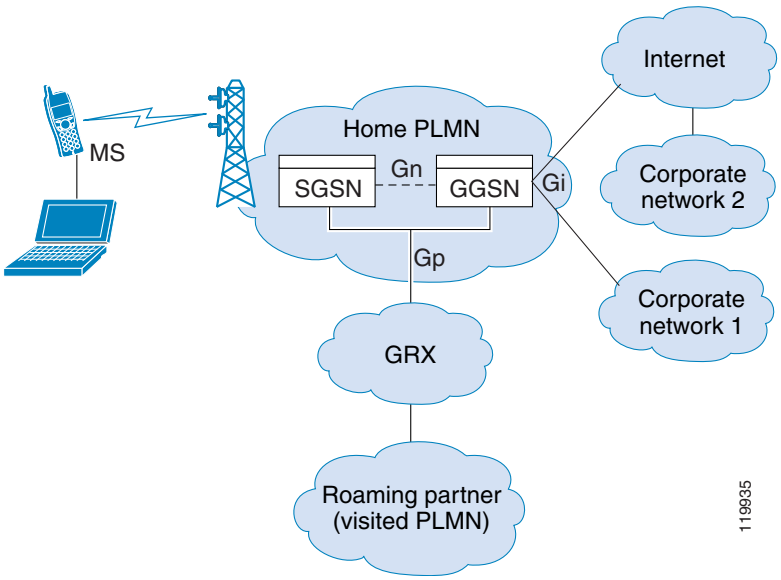
- GTP Inspection Overview, page 26-33
- Configuring a GTP Inspection Policy Map for Additional Inspection Control, page 26-34
- Verifying and Monitoring GTP Inspection, page 26-38

Note GTP inspection requires a special license. If you enter GTP-related commands on a security appliance without the required license, the security appliance displays an error message.

GTP Inspection Overview

GPRS provides uninterrupted connectivity for mobile subscribers between GSM networks and corporate networks or the Internet. The GGSN is the interface between the GPRS wireless data network and other networks. The SGSN performs mobility, data session management, and data compression (See Figure 26-3).

Figure 26-3 GPRS Tunneling Protocol



The UMTS is the commercial convergence of fixed-line telephony, mobile, Internet and computer technology. UTRAN is the networking protocol used for implementing wireless networks in this system. GTP allows multi-protocol packets to be tunneled through a UMTS/GPRS backbone between a GGSN, an SGSN and the UTRAN.

GTP does not include any inherent security or encryption of user data, but using GTP with the security appliance helps protect your network against these risks.

The SGSN is logically connected to a GGSN using GTP. GTP allows multiprotocol packets to be tunneled through the GPRS backbone between GSNs. GTP provides a tunnel control and management protocol that allows the SGSN to provide GPRS network access for a mobile station by creating, modifying, and deleting tunnels. GTP uses a tunneling mechanism to provide a service for carrying user data packets.

Note When using GTP with failover, if a GTP connection is established and the active unit fails before data is transmitted over the tunnel, the GTP data connection (with a "j" flag set) is not replicated to the standby unit. This occurs because the active unit does not replicate embryonic connections to the standby unit.

Configuring a GTP Inspection Policy Map for Additional Inspection Control

If you want to enforce additional parameters on GTP traffic, create and configure a GTP map. If you do not specify a map with the **inspect gtp** command, the security appliance uses the default GTP map, which is preconfigured with the following default values:

- **request-queue 200**
- **timeout gsn 0:30:00**
- **timeout pdp-context 0:30:00**
- **timeout request 0:01:00**
- **timeout signaling 0:30:00**
- **timeout tunnel 0:01:00**
- **tunnel-limit 500**

To create and configure a GTP map, perform the following steps. You can then apply the GTP map when you enable GTP inspection according to the "Configuring Application Inspection" section on page 26-5.

Step 1 Create a GTP inspection policy map, enter the following command:

```
hostname(config)# policy-map type inspect gtp policy_map_name
hostname(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 2 (Optional) To add a description to the policy map, enter the following command:

```
hostname(config-pmap)# description string
```

Step 3 To match an Access Point name, enter the following command:

```
hostname(config-pmap)# match [not] apn regex [regex_name | class regex_class_name]
```

Where the *regex_name* is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

Step 4 To match a message ID, enter the following command:

```
hostname(config-pmap)# match [not] message id [message_id | range lower_range upper_range]
```

Where the *message_id* is an alphanumeric identifier between 1 and 255. The *lower_range* is lower range of message IDs. The *upper_range* is the upper range of message IDs.

Step 5 To match a message length, enter the following command:

```
hostname(config-pmap)# match [not] message length min min_length max max_length
```

Where the *min_length* and *max_length* are both between 1 and 65536. The length specified by this command is the sum of the GTP header and the rest of the message, which is the payload of the UDP packet.

Step 6 To match the version, enter the following command:

```
hostname(config-pmap)# match [not] version [version_id | range lower_range upper_range]
```

Where the *version_id* is between 0and 255. The *lower_range* is lower range of versions. The *upper_range* is the upper range of versions.

Step 7 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

The **mnc** *network_code* argument is a two or three-digit value identifying the network code.

By default, the security appliance does not check for valid MCC/MNC combinations. This command is used for IMSI Prefix filtering. The MCC and MNC in the IMSI of the received packet is compared with the MCC/MNC configured with this command and is dropped if it does not match.

This command must be used to enable IMSI Prefix filtering. You can configure multiple instances to specify permitted MCC and MNC combinations. By default, the security appliance does not check the validity of MNC and MCC combinations, so you must verify the validity of the combinations configured. To find more information about MCC and MNC codes, see the ITU E.212 recommendation, *Identification Plan for Land Mobile Stations*.

b. To allow invalid GTP packets or packets that otherwise would fail parsing and be dropped, enter the following command:

```
hostname(config-pmap-p)# permit errors
```

By default, all invalid packets or packets that failed, during parsing, are dropped.

c. To enable support for GSN pooling, use the **permit response** command.

If the security appliance performs GTP inspection, by default the security appliance drops GTP responses from GSNs that were not specified in the GTP request. This situation occurs when you use load-balancing among a pool of GSNs to provide efficiency and scalability of GPRS.

You can enable support for GSN pooling by using the **permit response** command. This command configures the security appliance to allow responses from any of a designated set of GSNs, regardless of the GSN to which a GTP request was sent. You identify the pool of load-balancing GSNs as a network object. Likewise, you identify the SGSN as a network object. If the GSN responding belongs to the same object group as the GSN that the GTP request was sent to and if the SGSN is in a object group that the responding GSN is permitted to send a GTP response to, the security appliance permits the response.

d. To create an object to represent the pool of load-balancing GSNs, perform the following steps:

Use the **object-group** command to define a new network object group representing the pool of load-balancing GSNs.

```
hostname(config)# object-group network GSN-pool-name
hostname(config-network)#
```

For example, the following command creates an object group named gsnpool32:

```
hostname(config)# object-group network gsnpool32
hostname(config-network)#
```

e. Use the **network-object** command to specify the load-balancing GSNs. You can do so with one **network-object** command per GSN, using the **host** keyword. You can also using **network-object** command to identify whole networks containing GSNs that perform load balancing.

```
hostname(config-network)# network-object host IP-address
```

For example, the following commands create three network objects representing individual hosts:

```
hostname(config-network)# network-object host 192.168.100.1
hostname(config-network)# network-object host 192.168.100.2
hostname(config-network)# network-object host 192.168.100.3
hostname(config-network)#
```

f. To create an object to represent the SGSN that the load-balancing GSNs are permitted to respond to, perform the following steps:

a. Use the **object-group** command to define a new network object group that will represent the SGSN that sends GTP requests to the GSN pool.

```
hostname(config)# object-group network SGSN-name
hostname(config-network)#
```

For example, the following command creates an object group named sgsn32:

```
hostname(config)# object-group network sgsn32
hostname(config-network)#
```

b. Use the **network-object** command with the **host** keyword to identify the SGSN.

```
hostname(config-network)# network-object host IP-address
```

For example, the following command creates a network objects representing the SGSN:

```
hostname(config-network)# network-object host 192.168.50.100
hostname(config-network)#
```

g. To allow GTP responses from any GSN in the network object representing the GSN pool, defined in c., d, to the network object representing the SGSN, defined in c., f., enter the following commands:

```
hostname(config)# gtp-map map_name
hostname(config-gtp-map)# permit response to-object-group SGSN-name from-object-group
GSN-pool-name
```

For example, the following command permits GTP responses from any host in the object group named gsnpool32 to the host in the object group named sgsn32:

```
hostname(config-gtp-map)# permit response to-object-group sgsn32 from-object-group
gsnpool32
```

The following example shows how to support GSN pooling by defining network objects for the GSN pool and the SGSN. An entire Class C network is defined as the GSN pool but you can identify multiple individual IP addresses, one per **network-object** command, instead of identifying whole networks. The example then modifies a GTP map to permit responses from the GSN pool to the SGSN.

```
hostname(config)# object-group network gsnpool32
hostname(config-network)# network-object 192.168.100.0 255.255.255.0
hostname(config)# object-group network sgsn32
hostname(config-network)# network-object host 192.168.50.100
hostname(config)# gtp-map gtp-policy
hostname(config-gtp-map)# permit response to-object-group sgsn32 from-object-group
gsnpool32
```

h. To specify the maximum number of GTP requests that will be queued waiting for a response, enter the following command:

```
hostname(config-gtp-map)# request-queue max_requests
```

where the *max_requests* argument sets the maximum number of GTP requests that will be queued waiting for a response, from 1 to 4294967295. The default is 200.

When the limit has been reached and a new request arrives, the request that has been in the queue for the longest time is removed. The Error Indication, the Version Not Supported and the SGSN Context Acknowledge messages are not considered as requests and do not enter the request queue to wait for a response.

i. To change the inactivity timers for a GTP session, enter the following command:

```
hostname(config-gtp-map)# timeout {gsn | pdp-context | request | signaling | tunnel}
hh:mm:ss
```

Enter this command separately for each timeout.

The **gsn** keyword specifies the period of inactivity after which a GSN will be removed.

The **pdp-context** keyword specifies the maximum period of time allowed before beginning to receive the PDP context.

The **request** keyword specifies the maximum period of time allowed before beginning to receive the GTP message.

The **signaling** keyword specifies the period of inactivity after which the GTP signaling will be removed.

The **tunnel** keyword specifies the period of inactivity after which the GTP tunnel will be torn down.

The *hh*:*mm*:*ss* argument is the timeout where *hh* specifies the hour, *mm* specifies the minutes, and *ss* specifies the seconds. The value **0** means never tear down.

j. To specify the maximum number of GTP tunnels allowed to be active on the security appliance, enter the following command:

```
hostname(config-gtp-map)# tunnel-limit max_tunnels
```

where the *max_tunnels* argument is the maximum number of tunnels allowed, from 1 to 4294967295. The default is 500.

New requests will be dropped once the number of tunnels specified by this command is reached.

The following example shows how to limit the number of tunnels in the network:

```
hostname(config)# policy-map type inspect gtp gmap
hostname(config-pmap)# parameters
hostname(config-pmap-p)# tunnel-limit 3000

hostname(config)# policy-map global_policy
hostname(config-pmap)# class inspection_default
hostname(config-pmap-c)# inspect gtp gmap

hostname(config)# service-policy global_policy global
```

Verifying and Monitoring GTP Inspection

To display GTP configuration, enter the **show service-policy inspect gtp** command in privileged EXEC mode. For the detailed syntax for this command, see the command page in the *Cisco Security Appliance Command Reference*.

Use the **show service-policy inspect gtp statistics** command to show the statistics for GTP inspection. The following is sample output from the **show service-policy inspect gtp statistics** command:

```
hostname# show service-policy inspect gtp statistics
GPRS GTP Statistics:
  version_not_support               0     msg_too_short             0
  unknown_msg                       0     unexpected_sig_msg        0
  unexpected_data_msg               0     ie_duplicated             0
  mandatory_ie_missing              0     mandatory_ie_incorrect    0
  optional_ie_incorrect             0     ie_unknown                0
  ie_out_of_order                   0     ie_unexpected             0
  total_forwarded                   0     total_dropped             0
  signalling_msg_dropped            0     data_msg_dropped          0
  signalling_msg_forwarded          0     data_msg_forwarded        0
  total created_pdp                 0     total deleted_pdp         0
  total created_pdpmcb              0     total deleted_pdpmcb      0
  pdp_non_existent                  0
```

You can use the vertical bar (|) to filter the display. Type **?|** for more display filtering options.

The following is sample GSN output from the **show service-policy inspect gtp statistics gsn** command:

```
hostname# show service-policy inspect gtp statistics gsn 9.9.9.9
1 in use, 1 most used, timeout 0:00:00

GTP GSN Statistics for 9.9.9.9, Idle 0:00:00, restart counter 0
    Tunnels Active 0Tunnels Created 0
    Tunnels Destroyed 0
    Total Messages Received 2
    Signaling Messages Data Messages
    total received 2 0
    dropped 0 0
    forwarded 2 0
```

Use the **show service-policy inspect gtp pdp-context** command to display PDP context-related information. The following is sample output from the **show service-policy inspect gtp pdp-context** command:

```
hostname# show service-policy inspect gtp pdp-context detail
1 in use, 1 most used, timeout 0:00:00

Version TID                 MS Addr        SGSN Addr    Idle      APN
v1      1234567890123425         10.0.1.1        10.0.0.2 0:00:13  gprs.cisco.com

    user_name (IMSI): 214365870921435    MS address:         1.1.1.1
```

```
primary pdp: Y                          nsapi: 2
sgsn_addr_signal:        10.0.0.2       sgsn_addr_data:          10.0.0.2
ggsn_addr_signal:        10.1.1.1       ggsn_addr_data:          10.1.1.1
sgsn control teid:     0x000001d1       sgsn data teid:        0x000001d3
ggsn control teid:     0x6306ffa0       ggsn data teid:        0x6305f9fc
seq_tpdu_up:                  0       seq_tpdu_down:                  0
signal_sequence:              0
upstream_signal_flow:         0       upstream_data_flow:             0
downstream_signal_flow:       0       downstream_data_flow:           0
RAupdate_flow:                0
```

The PDP context is identified by the tunnel ID, which is a combination of the values for IMSI and NSAPI. A GTP tunnel is defined by two associated PDP contexts in different GSN nodes and is identified with a Tunnel ID. A GTP tunnel is necessary to forward packets between an external packet data network and a MS user.

You can use the vertical bar (|) to filter the display, as in the following example:

```
hostname# show service-policy gtp statistics  |  grep gsn
```

H.323 Inspection

This section describes the H.323 application inspection. This section includes the following topics:

- H.323 Inspection Overview, page 26-39
- How H.323 Works, page 26-40
- Limitations and Restrictions, page 26-41
- Configuring H.323 and H.225 Timeout Values, page 26-44
- Verifying and Monitoring H.323 Inspection, page 26-44

H.323 Inspection Overview

H.323 inspection provides support for H.323 compliant applications such as Cisco CallManager and VocalTec Gatekeeper. H.323 is a suite of protocols defined by the International Telecommunication Union for multimedia conferences over LANs. The security appliance supports H.323 through Version 4, including H.323 v3 feature Multiple Calls on One Call Signaling Channel.

With H323 inspection enabled, the security appliance supports multiple calls on the same call signaling channel, a feature introduced with H.323 Version 3. This feature reduces call setup time and reduces the use of ports on the security appliance.

The two major functions of H.323 inspection are as follows:

- NAT the necessary embedded IPv4 addresses in the H.225 and H.245 messages. Because H.323 messages are encoded in PER encoding format, the security appliance uses an ASN.1 decoder to decode the H.323 messages.
- Dynamically allocate the negotiated H.245 and RTP/RTCP connections.

How H.323 Works

The H.323 collection of protocols collectively may use up to two TCP connection and four to six UDP connections. FastConnect uses only one TCP connection, and RAS uses a single UDP connection for registration, admissions, and status.

An H.323 client may initially establish a TCP connection to an H.323 server using TCP port 1720 to request Q.931 call setup. As part of the call setup process, the H.323 terminal supplies a port number to the client to use for an H.245 TCP connection. In environments where H.323 gatekeeper is in use, the initial packet is transmitted using UDP.

H.323 inspection monitors the Q.931 TCP connection to determine the H.245 port number. If the H.323 terminals are not using FastConnect, the security appliance dynamically allocates the H.245 connection based on the inspection of the H.225 messages.

Within each H.245 message, the H.323 endpoints exchange port numbers that are used for subsequent UDP data streams. H.323 inspection inspects the H.245 messages to identify these ports and dynamically creates connections for the media exchange. RTP uses the negotiated port number, while RTCP uses the next higher port number.

The H.323 control channel handles H.225 and H.245 and H.323 RAS. H.323 inspection uses the following ports.

- 1718—Gate Keeper Discovery UDP port
- 1719—RAS UDP port
- 1720—TCP Control Port

You must permit traffic for the well-known H.323 port 1720 for the H.225 call signaling; however, the H.245 signaling ports are negotiated between the endpoints in the H.225 signaling. When an H.323 gatekeeper is used, the security appliance opens an H.225 connection based on inspection of the ACF message.

After inspecting the H.225 messages, the security appliance opens the H.245 channel and then inspects traffic sent over the H.245 channel as well. All H.245 messages passing through the security appliance undergo H.245 application inspection, which translates embedded IP addresses and opens the media channels negotiated in H.245 messages.

The H.323 ITU standard requires that a TPKT header, defining the length of the message, precede the H.225 and H.245, before being passed on to the reliable connection. Because the TPKT header does not necessarily need to be sent in the same TCP packet as H.225 and H.245 messages, the security appliance must remember the TPKT length to process and decode the messages properly. For each connection, the security appliance keeps a record that contains the TPKT length for the next expected message.

If the security appliance needs to perform NAT on IP addresses in messages, it changes the checksum, the UUIE length, and the TPKT, if it is included in the TCP packet with the H.225 message. If the TPKT is sent in a separate TCP packet, the security appliance proxy ACKs that TPKT and appends a new TPKT to the H.245 message with the new length.

Note The security appliance does not support TCP options in the Proxy ACK for the TPKT.

Each UDP connection with a packet going through H.323 inspection is marked as an H.323 connection and times out with the H.323 timeout as configured with the **timeout** command.

Limitations and Restrictions

The following are some of the known issues and limitations when using H.323 application inspection:

- Static PAT may not properly translate IP addresses embedded in optional fields within H.323 messages. If you experience this kind of problem, do not use static PAT with H.323.
- H.323 application inspection is not supported with NAT between same-security-level interfaces.
- When a NetMeeting client registers with an H.323 gatekeeper and tries to call an H.323 gateway that is also registered with the H.323 gatekeeper, the connection is established but no voice is heard in either direction. This problem is unrelated to the security appliance.
- If you configure a network static address where the network static address is the same as a third-party netmask and address, then any outbound H.323 connection fails.

Configuring an H.323 Inspection Policy Map for Additional Inspection Control

To specify actions when a message violates a parameter, create an H.323 inspection policy map. You can then apply the inspection policy map when you enable H.323 inspection according to the "Configuring Application Inspection" section on page 26-5.

To create an H.323 inspection policy map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the "Creating a Regular Expression" section on page 16-13. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the "Creating a Regular Expression Class Map" section on page 16-16.s

Step 3 (Optional) Create an H.323 inspection class map by performing the following steps.

A class map groups multiple traffic matches. Traffic must match *all* of the **match** commands to match the class map. You can alternatively identify **match** commands directly in the policy map. The difference between creating a class map and defining the traffic match directly in the inspection policy map is that the class map lets you create more complex match criteria, and you can reuse class maps.

To specify traffic that should not match the class map, use the **match not** command. For example, if the **match not** command specifies the string "example.com," then any traffic that includes "example.com" does not match the class map.

For the traffic that you identify in this class map, you can specify actions such as drop-connection, reset, and/or log the connection in the inspection policy map.

If you want to perform different actions for each **match** command, you should identify the traffic directly in the policy map.

a. Create the class map by entering the following command:

```
hostname(config)# class-map type inspect h323 [match-all | match-any] class_map_name
hostname(config-cmap)#
```

Where *the class_map_name* is the name of the class map. The **match-all** keyword is the default, and specifies that traffic must match all criteria to match the class map. The **match-any** keyword specifies that the traffic matches the class map if it matches at least one of the criteria. The CLI enters class-map configuration mode, where you can enter one or more **match** commands.

b. (Optional) To add a description to the class map, enter the following command:

```
hostname(config-cmap)# description string
```

Where *string* is the description of the class map (up to 200 characters).

c. (Optional) To match a called party, enter the following command:

```
hostname(config-cmap)# match [not] called-party regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

d. (Optional) To match a media type, enter the following command:

```
hostname(config-cmap)# match [not] media-type {audio | data | video}
```

Step 4 Create an H.323 inspection policy map, enter the following command:

```
hostname(config)# policy-map type inspect h323 policy_map_name
hostname(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 5 (Optional) To add a description to the policy map, enter the following command:

```
hostname(config-pmap)# description string
```

Step 6 To apply actions to matching traffic, perform the following steps.

a. Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the H.323 class map that you created in Step 3 by entering the following command:

  ```
  hostname(config-pmap)# class class_map_name
  hostname(config-pmap-c)#
  ```

- Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
hostname(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

Not all options are available for each **match** or **class** command. See the CLI help or the *Cisco Security Appliance Command Reference* for the exact options available.

The **drop** keyword drops all packets that match.

The **send-protocol-error** keyword sends a protocol error message.

The **drop-connection** keyword drops the packet and closes the connection.

The **mask** keyword masks out the matching portion of the packet.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

The **rate-limit** *message_rate* argument limits the rate of messages.

You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 16-9.

Step 7 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

b. To define the H.323 call duration limit, enter the following command:

```
hostname(config-pmap-p)# call-duration-limit time
```

Where *time* is the call duration limit in seconds. Range is from 0:0:0 ti 1163:0;0. A value of 0 means never timeout.

c. To enforce call party number used in call setup, enter the following command:

```
hostname(config-pmap-p)# call-party-number
```

d. To enforce H.245 tunnel blocking, enter the following command:

```
hostname(config-pmap-p)# h245-tunnel-block action {drop-connection | log}
```

e. To define an hsi group and enter hsi group configuration mode, enter the following command:

```
hostname(config-pmap-p)# hsi-group id
```

Where *id* is the hsi group ID. Range is from 0 to 2147483647.

To add an hsi to the hsi group, enter the following command in hsi group configuration mode:

```
hostname(config-h225-map-hsi-grp)# hsi ip_address
```

Where *ip_address* is the host to add. A maximum of five hosts per hsi group are allowed.

To add an endpoint to the hsi group, enter the following command in hsi group configuration mode:

```
hostname(config-h225-map-hsi-grp)# endpoint ip_address if_name
```

Where *ip_address* is the endpoint to add and *if_name* is the interface through which the endpoint is connected to the security appliance. A maximum of ten endpoints per hsi group are allowed.

f. To check RTP packets flowing on the pinholes for protocol conformance, enter the following command:

```
hostname(config-pmap-p)# rtp-conformance [enforce-payloadtype]
```

Where the **enforce-payloadtype** keyword enforces the payload type to be audio or video based on the signaling exchange.

g. To enable state checking validation, enter the following command:

```
hostname(config-pmap-p)# state-checking {h225 | ras}
```

The following example shows how to configure phone number filtering:

```
hostname(config)# regex caller 1 "5551234567"
hostname(config)# regex caller 2 "5552345678"
hostname(config)# regex caller 3 "5553456789"

hostname(config)# class-map type inspect h323 match-all h323_traffic
hostname(config-pmap-c)# match called-party regex caller1
hostname(config-pmap-c)# match calling-party regex caller2

hostname(config)# policy-map type inspect h323 h323_map
```

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)# class h323_traffic
hostname(config-pmap-c)# drop
```

Configuring H.323 and H.225 Timeout Values

To configure the idle time after which an H.225 signalling connection is closed, use the **timeout h225** command. The default for H.225 timeout is one hour.

To configure the idle time after which an H.323 control connection is closed, use the **timeout h323** command. The default is five minutes.

Verifying and Monitoring H.323 Inspection

This section describes how to display information about H.323 sessions. This section includes the following topics:

- Monitoring H.225 Sessions, page 26-44
- Monitoring H.245 Sessions, page 26-45
- Monitoring H.323 RAS Sessions, page 26-45

Monitoring H.225 Sessions

The **show h225** command displays information for H.225 sessions established across the security appliance. Along with the **debug h323 h225 event**, **debug h323 h245 event**, and **show local-host** commands, this command is used for troubleshooting H.323 inspection engine issues.

Before entering the **show h225**, **show h245**, or **show h323-ras** commands, we recommend that you configure the **pager** command. If there are a lot of session records and the **pager** command is not configured, it may take a while for the **show** command output to reach its end. If there is an abnormally large number of connections, check that the sessions are timing out based on the default timeout values or the values set by you. If they are not, then there is a problem that needs to be investigated.

The following is sample output from the **show h225** command:

```
hostname# show h225
Total H.323 Calls: 1
1 Concurrent Call(s) for
    Local:   10.130.56.3/1040   Foreign: 172.30.254.203/1720
    1. CRV 9861
    Local:   10.130.56.3/1040   Foreign: 172.30.254.203/1720
0 Concurrent Call(s) for
    Local:   10.130.56.4/1050   Foreign: 172.30.254.205/1720
```

This output indicates that there is currently 1 active H.323 call going through the security appliance between the local endpoint 10.130.56.3 and foreign host 172.30.254.203, and for these particular endpoints, there is 1 concurrent call between them, with a CRV for that call of 9861.

For the local endpoint 10.130.56.4 and foreign host 172.30.254.205, there are 0 concurrent calls. This means that there is no active call between the endpoints even though the H.225 session still exists. This could happen if, at the time of the **show h225** command, the call has already ended but the H.225 session has not yet been deleted. Alternately, it could mean that the two endpoints still have a TCP connection

opened between them because they set “maintainConnection” to TRUE, so the session is kept open until they set it to FALSE again, or until the session times out based on the H.225 timeout value in your configuration.

Monitoring H.245 Sessions

The **show h245** command displays information for H.245 sessions established across the security appliance by endpoints using slow start. Slow start is when the two endpoints of a call open another TCP control channel for H.245. Fast start is where the H.245 messages are exchanged as part of the H.225 messages on the H.225 control channel.) Along with the **debug h323 h245 event**, **debug h323 h225 event**, and **show local-host** commands, this command is used for troubleshooting H.323 inspection engine issues.

The following is sample output from the **show h245** command:

```
hostname# show h245
Total: 1
        LOCAL           TPKT    FOREIGN         TPKT
1       10.130.56.3/1041        0       172.30.254.203/1245    0
        MEDIA: LCN 258 Foreign 172.30.254.203 RTP 49608 RTCP 49609
                       Local   10.130.56.3 RTP 49608 RTCP 49609
        MEDIA: LCN 259 Foreign 172.30.254.203 RTP 49606 RTCP 49607
                       Local   10.130.56.3 RTP 49606 RTCP 49607
```

There is currently one H.245 control session active across the security appliance. The local endpoint is 10.130.56.3, and we are expecting the next packet from this endpoint to have a TPKT header because the TPKT value is 0. The TKTP header is a 4-byte header preceding each H.225/H.245 message. It gives the length of the message, including the 4-byte header. The foreign host endpoint is 172.30.254.203, and we are expecting the next packet from this endpoint to have a TPKT header because the TPKT value is 0.

The media negotiated between these endpoints have an LCN of 258 with the foreign RTP IP address/port pair of 172.30.254.203/49608 and an RTCP IP address/port of 172.30.254.203/49609 with a local RTP IP address/port pair of 10.130.56.3/49608 and an RTCP port of 49609.

The second LCN of 259 has a foreign RTP IP address/port pair of 172.30.254.203/49606 and an RTCP IP address/port pair of 172.30.254.203/49607 with a local RTP IP address/port pair of 10.130.56.3/49606 and RTCP port of 49607.

Monitoring H.323 RAS Sessions

The **show h323-ras** command displays information for H.323 RAS sessions established across the security appliance between a gatekeeper and its H.323 endpoint. Along with the **debug h323 ras event** and **show local-host** commands, this command is used for troubleshooting H.323 RAS inspection engine issues.

The **show h323-ras** command displays connection information for troubleshooting H.323 inspection engine issues. The following is sample output from the **show h323-ras** command:

```
hostname# show h323-ras
Total: 1
        GK                      Caller
        172.30.254.214 10.130.56.14
```

This output shows that there is one active registration between the gatekeeper 172.30.254.214 and its client 10.130.56.14.

HTTP Inspection

This section describes the HTTP inspection engine. This section includes the following topics:

- HTTP Inspection Overview, page 26-46
- Configuring an HTTP Inspection Policy Map for Additional Inspection Control, page 26-46

HTTP Inspection Overview

Use the HTTP inspection engine to protect against specific attacks and other threats that may be associated with HTTP traffic. HTTP inspection performs several functions:

- Enhanced HTTP inspection
- URL screening through N2H2 or Websense
- Java and ActiveX filtering

The latter two features are configured in conjunction with the **filter** command. For more information about filtering, see Chapter 22, "Applying Filtering Services."

The enhanced HTTP inspection feature, which is also known as an application firewall and is available when you configure an HTTP map (see "Configuring an HTTP Inspection Policy Map for Additional Inspection Control"), can help prevent attackers from using HTTP messages for circumventing network security policy. It verifies the following for all HTTP messages:

- Conformance to RFC 2616
- Use of RFC-defined methods only.
- Compliance with the additional criteria.

Configuring an HTTP Inspection Policy Map for Additional Inspection Control

To specify actions when a message violates a parameter, create an HTTP inspection policy map. You can then apply the inspection policy map when you enable HTTP inspection according to the "Configuring Application Inspection" section on page 26-5.

Note When you enable HTTP inspection with an inspection policy map, strict HTTP inspection with the action reset and log is enabled by default. You can change the actions performed in response to inspection failure, but you cannot disable strict inspection as long as the inspection policy map remains enabled.

To create an HTTP inspection policy map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the "Creating a Regular Expression" section on page 16-13. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the "Creating a Regular Expression Class Map" section on page 16-16.

Step 3 (Optional) Create an HTTP inspection class map by performing the following steps.

A class map groups multiple traffic matches. Traffic must match *all* of the **match** commands to match the class map. You can alternatively identify **match** commands directly in the policy map. The difference between creating a class map and defining the traffic match directly in the inspection policy map is that the class map lets you create more complex match criteria, and you can reuse class maps.

To specify traffic that should not match the class map, use the **match not** command. For example, if the **match not** command specifies the string "example.com," then any traffic that includes "example.com" does not match the class map.

For the traffic that you identify in this class map, you can specify actions such as drop, drop-connection, reset, mask, set the rate limit, and/or log the connection in the inspection policy map.

If you want to perform different actions for each **match** command, you should identify the traffic directly in the policy map.

a. Create the class map by entering the following command:

```
hostname(config)# class-map type inspect http [match-all | match-any] class_map_name
hostname(config-cmap)#
```

Where *class_map_name* is the name of the class map. The **match-all** keyword is the default, and specifies that traffic must match all criteria to match the class map. The **match-any** keyword specifies that the traffic matches the class map if it matches at least one of the criteria. The CLI enters class-map configuration mode, where you can enter one or more **match** commands.

b. (Optional) To add a description to the class map, enter the following command:

```
hostname(config-cmap)# description string
```

c. (Optional) To match traffic with a content-type field in the HTTP response that does not match the accept field in the corresponding HTTP request message, enter the following command:

```
hostname(config-cmap)# match [not] req-resp content-type mismatch
```

d. (Optional) To match text found in the HTTP request message arguments, enter the following command:

```
hostname(config-cmap)# match [not] request args regex [regex_name | class
regex_class_name]
```

Where the *regex_name* is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

e. (Optional) To match text found in the HTTP request message body or to match traffic that exceeds the maximum HTTP request message body length, enter the following command:

```
hostname(config-cmap)# match [not] request body {regex [regex_name | class
regex_class_name] | length gt max_bytes}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2. The **length gt** *max_bytes* is the maximum message body length in bytes.

f. (Optional) To match text found in the HTTP request message header, or to restrict the count or length of the header, enter the following command:

```
hostname(config-cmap)# match [not] request header {[field]
[regex [regex_name | class regex_class_name]] |
[length gt max_length_bytes | count gt max_count_bytes]}
```

Where the *field* is the predefined message header keyword. The **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2. The **length gt** *max_bytes* is the maximum message body length in bytes. The **count gt** *max_count* is the maximum number of header fields.

g. (Optional) To match text found in the HTTP request message method, enter the following command:

```
hostname(config-cmap)# match [not] request method {[method] |
[regex [regex_name | class regex_class_name]]
```

Where the *method* is the predefined message method keyword. The **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

h. (Optional) To match text found in the HTTP request message URI, enter the following command:

```
hostname(config-cmap)# match [not] request uri {regex [regex_name | class
regex_class_name] | length gt max_bytes}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2. The **length gt** *max_bytes* is the maximum message body length in bytes.

i. Optional) To match text found in the HTTP response message body, or to comment out Java applet and Active X object tags in order to filter them, enter the following command:

```
hostname(config-cmap)# match [not] response body {[active-x] | [java-applet] |
[regex [regex_name | class regex_class_name]] | length gt max_bytes}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2. The **length gt** *max_bytes* is the maximum message body length in bytes.

j. (Optional) To match text found in the HTTP response message header, or to restrict the count or length of the header, enter the following command:

```
hostname(config-cmap)# match [not] response header {[field]
[regex [regex_name | class regex_class_name]] |
[length gt max_length_bytes | count gt max_count]}
```

Where the *field* is the predefined message header keyword. The **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2. The **length gt** *max_bytes* is the maximum message body length in bytes. The **count gt** *max_count* is the maximum number of header fields.

k. (Optional) To match text found in the HTTP response message status line, enter the following command:

```
hostname(config-cmap)# match [not] response status-line {regex [regex_name | class
regex_class_name]}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

Step 4 Create an HTTP inspection policy map, enter the following command:

```
hostname(config)# policy-map type inspect http policy_map_name
hostname(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 5 (Optional) To add a description to the policy map, enter the following command:

```
hostname(config-pmap)# description string
```

Step 6 To apply actions to matching traffic, perform the following steps.

a. Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the HTTP class map that you created in Step 3 by entering the following command:

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

- Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
hostname(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

Not all options are available for each **match** or **class** command. See the CLI help or the *Cisco Security Appliance Command Reference* for the exact options available.

The **drop** keyword drops all packets that match.

The **send-protocol-error** keyword sends a protocol error message.

The **drop-connection** keyword drops the packet and closes the connection.

The **mask** keyword masks out the matching portion of the packet.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

The **rate-limit** *message_rate* argument limits the rate of messages.

You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 16-9.

Step 7 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

b. To check for HTTP protocol violations, enter the following command:

```
hostname(config-pmap-p)# protocol-violation [action [drop-connection | reset | log]]
```

Where the **drop-connection** action closes the connection. The **reset** action closes the connection and sends a TCP reset to the client. The **log** action sends a system log message when this policy map matches traffic.

c. To substitute a string for the server header field, enter the following command:

```
hostname(config-pmap-p)# spoof-server string
```

Where the *string* argument is the string to substitute for the server header field. Note: WebVPN streams are not subject to the **spoof-server** comand.

The following example shows how to define an HTTP inspection policy map that will allow and log any HTTP connection that attempts to access “www\.xyz.com/.*\.asp" or "www\.xyz[0-9][0-9]\.com" with methods "GET" or "PUT." All other URL/Method combinations will be silently allowed.

```
hostname(config)# regex url1 "www\.xyz.com/.*\.asp"
hostname(config)# regex url2 "www\.xyz[0-9][0-9]\.com"
hostname(config)# regex get "GET"
hostname(config)# regex put "PUT"

hostname(config)# class-map type regex match-any url_to_log
hostname(config-cmap)# match regex url1
hostname(config-cmap)# match regex url2
hostname(config-cmap)# exit

hostname(config)# class-map type regex match-any methods_to_log
hostname(config-cmap)# match regex get
hostname(config-cmap)# match regex put
hostname(config-cmap)# exit

hostname(config)# class-map type inspect http http_url_policy
hostname(config-cmap)# match request uri regex class url_to_log
hostname(config-cmap)# match request method regex class methods_to_log
hostname(config-cmap)# exit

hostname(config)# policy-map type inspect http http_policy
hostname(config-pmap)# class http_url_policy
hostname(config-pmap-c)# log
```

Instant Messaging Inspection

This section describes the IM inspection engine. This section includes the following topics:

- IM Inspection Overview, page 26-50
- Configuring an Instant Messaging Inspection Policy Map for Additional Inspection Control, page 26-50

IM Inspection Overview

The IM inspect engine lets you apply fine grained controls on the IM application to control the network usage and stop leakage of confidential data, propagation of worms, and other threats to the corporate network.

Configuring an Instant Messaging Inspection Policy Map for Additional Inspection Control

To specify actions when a message violates a parameter, create an IM inspection policy map. You can then apply the inspection policy map when you enable IM inspection according to the “Configuring Application Inspection” section on page 26-5.

To create an IM inspection policy map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the "Creating a Regular Expression" section on page 16-13. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the "Creating a Regular Expression Class Map" section on page 16-16.s

Step 3 (Optional) Create an IM inspection class map by performing the following steps.

A class map groups multiple traffic matches. Traffic must match *all* of the **match** commands to match the class map. You can alternatively identify **match** commands directly in the policy map. The difference between creating a class map and defining the traffic match directly in the inspection policy map is that the class map lets you create more complex match criteria, and you can reuse class maps.

To specify traffic that should not match the class map, use the **match not** command. For example, if the **match not** command specifies the string "example.com," then any traffic that includes "example.com" does not match the class map.

For the traffic that you identify in this class map, you can specify actions such as drop-connection, reset, and/or log the connection in the inspection policy map.

If you want to perform different actions for each **match** command, you should identify the traffic directly in the policy map.

a. Create the class map by entering the following command:

```
hostname(config)# class-map type inspect im [match-all | match-any] class_map_name
hostname(config-cmap)#
```

Where *the class_map_name* is the name of the class map. The **match-all** keyword is the default, and specifies that traffic must match all criteria to match the class map. The **match-any** keyword specifies that the traffic matches the class map if it matches at least one of the criteria. The CLI enters class-map configuration mode, where you can enter one or more **match** commands.

b. (Optional) To add a description to the class map, enter the following command:

```
hostname(config-cmap)# description string
```

Where *the string* is the description of the class map (up to 200 characters).

c. (Optional) To match traffic of a specific IM protocol, such as Yahoo or MSN, enter the following command:

```
hostname(config-cmap)# match [not] protocol {im-yahoo | im-msn}
```

d. (Optional) To match a specific IM service, such as chat, file-transfer, webcam, voice-chat, conference, or games, enter the following command:

```
hostname(config-cmap)# match [not] service {chat | file-transfer | webcam | voice-chat
| conference | games}
```

e. (Optional) To match the source login name of the IM message, enter the following command:

```
hostname(config-cmap)# match [not] login-name regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

f. (Optional) To match the destination login name of the IM message, enter the following command:

```
hostname(config-cmap)# match [not] peer-login-name regex {class class_name |
regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

g. (Optional) To match the source IP address of the IM message, enter the following command:

```
hostname(config-cmap)# match [not] ip-address ip_address ip_address_mask
```

Where the *ip_address* and the *ip_address_mask* is the IP address and netmask of the message source.

h. (Optional) To match the destination IP address of the IM message, enter the following command:

```
hostname(config-cmap)# match [not] peer-ip-address ip_address ip_address_mask
```

Where the *ip_address* and the *ip_address_mask* is the IP address and netmask of the message destination.

i. (Optional) To match the version of the IM message, enter the following command:

```
hostname(config-cmap)# match [not] version regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

j. (Optional) To match the filename of the IM message, enter the following command:

```
hostname(config-cmap)# match [not] filename regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

Note Not supported using MSN IM protocol.

Step 4 Create an IM inspection policy map, enter the following command:

```
hostname(config)# policy-map type inspect im policy_map_name
hostname(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 5 (Optional) To add a description to the policy map, enter the following command:

```
hostname(config-pmap)# description string
```

Step 6 Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the IM class map that you created in Step 3 by entering the following command:

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

- Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 16-9.

Step 7 Specify the action you want to perform on the matching traffic by entering the following command:

```
hostname(config-pmap-c)# {drop-connection | reset | log}
```

Where the **drop-connection** action closes the connection. The **reset** action closes the connection and sends a TCP reset to the client. The **log** action sends a system log message when this policy map matches traffic.

The following example shows how to define an IM inspection policy map.

```
hostname(config)# regex loginname1 "ying\@yahoo.com"
hostname(config)# regex loginname2 "Kevin\@yahoo.com"
hostname(config)# regex loginname3 "rahul\@yahoo.com"
hostname(config)# regex loginname4 "darshant\@yahoo.com"
hostname(config)# regex yahoo_version_regex "1\.0"
hostname(config)# regex gif_files ".*\.gif"
hostname(config)# regex exe_files ".*\.exe"

hostname(config)# class-map type regex match-any yahoo_src_login_name_regex
hostname(config-cmap)# match regex loginname1
hostname(config-cmap)# match regex loginname2

hostname(config)# class-map type regex match-any yahoo_dst_login_name_regex
hostname(config-cmap)# match regex loginname3
hostname(config-cmap)# match regex loginname4

hostname(config)# class-map type inspect im match-any yahoo_file_block_list
hostname(config-cmap)# match filename regex gif_files
hostname(config-cmap)# match filename regex exe_files

hostname(config)# class-map type inspect im match-all yahoo_im_policy
hostname(config-cmap)# match login-name regex class yahoo_src_login_name_regex
hostname(config-cmap)# match peer-login-name regex class yahoo_dst_login_name_regex

hostname(config)# class-map type inspect im match-all yahoo_im_policy2
hostname(config-cmap)# match version regex yahoo_version_regex

hostname(config)# class-map im_inspect_class_map
hostname(config-cmap)# match default-inspection-traffic

hostname(config)# policy-map type inspect im im_policy_all
hostname(config-pmap)# class yahoo_file_block_list
hostname(config-pmap-c)# match service file-transfer
hostname(config-pmap)# class yahoo_im_policy
hostname(config-pmap-c)# drop-connection
hostname(config-pmap)# class yahoo_im_policy2
hostname(config-pmap-c)# reset
hostname(config)# policy-map global_policy_name
hostname(config-pmap)# class im_inspect_class_map
hostname(config-pmap-c)# inspect im im_policy_all
```

ICMP Inspection

The ICMP inspection engine allows ICMP traffic to have a "session" so it can be inspected like TCP and UDP traffic. Without the ICMP inspection engine, we recommend that you do not allow ICMP through the security appliance in an access list. Without stateful inspection, ICMP can be used to attack your network. The ICMP inspection engine ensures that there is only one response for each request, and that the sequence number is correct.

ICMP Error Inspection

When this feature is enabled, the security appliance creates translation sessions for intermediate hops that send ICMP error messages, based on the NAT configuration. The security appliance overwrites the packet with the translated IP addresses.

When disabled, the security appliance does not create translation sessions for intermediate nodes that generate ICMP error messages. ICMP error messages generated by the intermediate nodes between the inside host and the security appliance reach the outside host without consuming any additional NAT resource. This is undesirable when an outside host uses the traceroute command to trace the hops to the destination on the inside of the security appliance. When the security appliance does not translate the intermediate hops, all the intermediate hops appear with the mapped destination IP address.

The ICMP payload is scanned to retrieve the five-tuple from the original packet. Using the retrieved five-tuple, a lookup is performed to determine the original address of the client. The ICMP error inspection engine makes the following changes to the ICMP packet:

- In the IP Header, the mapped IP is changed to the real IP (Destination Address) and the IP checksum is modified.
- In the ICMP Header, the ICMP checksum is modified due to the changes in the ICMP packet.
- In the Payload, the following changes are made:
  - Original packet mapped IP is changed to the real IP
  - Original packet mapped port is changed to the real Port
  - Original packet IP checksum is recalculated

ILS Inspection

The ILS inspection engine provides NAT support for Microsoft NetMeeting, SiteServer, and Active Directory products that use LDAP to exchange directory information with an ILS server.

The security appliance supports NAT for ILS, which is used to register and locate endpoints in the ILS or SiteServer Directory. PAT cannot be supported because only IP addresses are stored by an LDAP database.

For search responses, when the LDAP server is located outside, NAT should be considered to allow internal peers to communicate locally while registered to external LDAP servers. For such search responses, xlates are searched first, and then DNAT entries to obtain the correct address. If both of these searches fail, then the address is not changed. For sites using NAT 0 (no NAT) and not expecting DNAT interaction, we recommend that the inspection engine be turned off to provide better performance.

Additional configuration may be necessary when the ILS server is located inside the security appliance border. This would require a hole for outside clients to access the LDAP server on the specified port, typically TCP 389.

Because ILS traffic only occurs on the secondary UDP channel, the TCP connection is disconnected after the TCP inactivity interval. By default, this interval is 60 minutes and can be adjusted using the **timeout** command.

ILS/LDAP follows a client/server model with sessions handled over a single TCP connection. Depending on the client's actions, several of these sessions may be created.

During connection negotiation time, a BIND PDU is sent from the client to the server. Once a successful BIND RESPONSE from the server is received, other operational messages may be exchanged (such as ADD, DEL, SEARCH, or MODIFY) to perform operations on the ILS Directory. The ADD REQUEST

and SEARCH RESPONSE PDUs may contain IP addresses of NetMeeting peers, used by H.323 (SETUP and CONNECT messages) to establish the NetMeeting sessions. Microsoft NetMeeting v2.X and v3.X provides ILS support.

The ILS inspection performs the following operations:

- Decodes the LDAP REQUEST/RESPONSE PDUs using the BER decode functions
- Parses the LDAP packet
- Extracts IP addresses
- Translates IP addresses as necessary
- Encodes the PDU with translated addresses using BER encode functions
- Copies the newly encoded PDU back to the TCP packet
- Performs incremental TCP checksum and sequence number adjustment

ILS inspection has the following limitations:

- Referral requests and responses are not supported
- Users in multiple directories are not unified
- Single users having multiple identities in multiple directories cannot be recognized by NAT

Note Because H225 call signalling traffic only occurs on the secondary UDP channel, the TCP connection is disconnected after the interval specified by the TCP **timeout** command. By default, this interval is set at 60 minutes.

MGCP Inspection

This section describes MGCP application inspection. This section includes the following topics:

- MGCP Inspection Overview, page 26-55
- Configuring an MGCP Inspection Policy Map for Additional Inspection Control, page 26-57
- Configuring MGCP Timeout Values, page 26-58
- Verifying and Monitoring MGCP Inspection, page 26-58

MGCP Inspection Overview

MGCP is a master/slave protocol used to control media gateways from external call control elements called media gateway controllers or call agents. A media gateway is typically a network element that provides conversion between the audio signals carried on telephone circuits and data packets carried over the Internet or over other packet networks. Using NAT and PAT with MGCP lets you support a large number of devices on an internal network with a limited set of external (global) addresses. Examples of media gateways are:

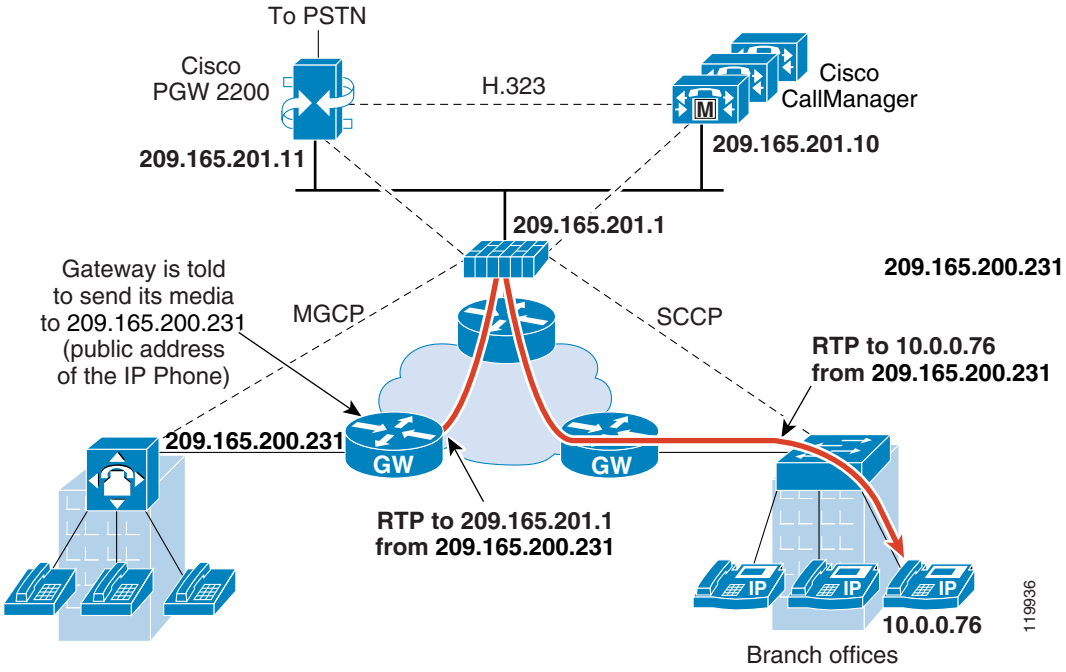
- Trunking gateways, that interface between the telephone network and a Voice over IP network. Such gateways typically manage a large number of digital circuits.
- Residential gateways, that provide a traditional analog (RJ11) interface to a Voice over IP network. Examples of residential gateways include cable modem/cable set-top boxes, xDSL devices, broad-band wireless devices.

- Business gateways, that provide a traditional digital PBX interface or an integrated soft PBX interface to a Voice over IP network.

Note To avoid policy failure when upgrading from ASA version 7.1, all layer 7 and layer 3 policies must have distinct names. For instance, a previously configured policy map with the same name as a previously configured MGCP map must be changed before the upgrade.

MGCP messages are transmitted over UDP. A response is sent back to the source address (IP address and UDP port number) of the command, but the response may not arrive from the same address as the command was sent to. This can happen when multiple call agents are being used in a failover configuration and the call agent that received the command has passed control to a backup call agent, which then sends the response. Figure 26-4 illustrates how NAT can be used with MGCP.

Figure 26-4 Using NAT with MGCP



MGCP endpoints are physical or virtual sources and destinations for data. Media gateways contain endpoints on which the call agent can create, modify and delete connections to establish and control media sessions with other multimedia endpoints. Also, the call agent can instruct the endpoints to detect certain events and generate signals. The endpoints automatically communicate changes in service state to the call agent.

MGCP transactions are composed of a command and a mandatory response. There are eight types of commands:

- CreateConnection
- ModifyConnection
- DeleteConnection
- NotificationRequest
- Notify

- AuditEndpoint
- AuditConnection
- RestartInProgress

The first four commands are sent by the call agent to the gateway. The Notify command is sent by the gateway to the call agent. The gateway may also send a DeleteConnection. The registration of the MGCP gateway with the call agent is achieved by the RestartInProgress command. The AuditEndpoint and the AuditConnection commands are sent by the call agent to the gateway.

All commands are composed of a Command header, optionally followed by a session description. All responses are composed of a Response header, optionally followed by a session description.

- The port on which the gateway receives commands from the call agent. Gateways usually listen to UDP port 2427.
- The port on which the call agent receives commands from the gateway. Call agents usually listen to UDP port 2727.

Note MGCP inspection does not support the use of different IP addresses for MGCP signaling and RTP data. A common and recommended practice is to send RTP data from a resilient IP address, such as a loopback or virtual IP address; however, the security appliance requires the RTP data to come from the same address as MGCP signalling.

Configuring an MGCP Inspection Policy Map for Additional Inspection Control

If the network has multiple call agents and gateways for which the security appliance has to open pinholes, create an MGCP map. You can then apply the MGCP map when you enable MGCP inspection according to the "Configuring Application Inspection" section on page 26-5

To create an MGCP map, perform the following steps:

Step 1 To create an MGCP inspection policy map, enter the following command:

```
hostname(config)# policy-map type inspect mgcp map_name
hostname(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 2 (Optional) To add a description to the policy map, enter the following command:

```
hostname(config-pmap)# description string
```

Step 3 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

b. To configure the call agents, enter the following command for each call agent:

```
hostname(config-pmap-p)# call-agent ip_address group_id
```

Use the **call-agent** command to specify a group of call agents that can manage one or more gateways. The call agent group information is used to open connections for the call agents in the group (other than the one a gateway sends a command to) so that any of the call agents can send the response.

call agents with the same *group_id* belong to the same group. A call agent may belong to more than one group. The *group_id* option is a number from 0 to 4294967295. The *ip_address* option specifies the IP address of the call agent.

Note MGCP call agents send AUEP messages to determine if MGCP end points are present. This establishes a flow through the security appliance and allows MGCP end points to register with the call agent.

c. To configure the gateways, enter the following command for each gateway:

```
hostname(config-pmap-p)# gateway ip_address group_id
```

Use the **gateway** command to specify which group of call agents are managing a particular gateway. The IP address of the gateway is specified with the *ip_address* option. The *group_id* option is a number from 0 to 4294967295 that must correspond with the *group_id* of the call agents that are managing the gateway. A gateway may only belong to one group.

d. If you want to change the maximum number of commands allowed in the MGCP command queue, enter the following command:

```
hostname(config-pmap-p)# command-queue command_limit
```

The following example shows how to define an MGCP map:

```
hostname(config)# policy-map type inspect mgcp sample_map
hostname(config-pmap)# parameters
hostname(config-pmap-p)# call-agent 10.10.11.5 101
hostname(config-pmap-p)# call-agent 10.10.11.6 101
hostname(config-pmap-p)# call-agent 10.10.11.7 102
hostname(config-pmap-p)# call-agent 10.10.11.8 102
hostname(config-pmap-p)# gateway 10.10.10.115 101
hostname(config-pmap-p)# gateway 10.10.10.116 102
hostname(config-pmap-p)# gateway 10.10.10.117 102
hostname(config-pmap-p)# command-queue 150
```

Configuring MGCP Timeout Values

The **timeout mgcp command** lets you set the interval for inactivity after which an MGCP media connection is closed. The default is 5 minutes.

The **timeout mgcp-pat** command lets you set the timeout for PAT xlates. Because MGCP does not have a keepalive mechanism, if you use non-Cisco MGCP gateways (call agents), the PAT xlates are torn down after the default timeout interval, which is 30 seconds.

Verifying and Monitoring MGCP Inspection

The **show mgcp commands** command lists the number of MGCP commands in the command queue. The **show mgcp sessions** command lists the number of existing MGCP sessions. The **detail** option includes additional information about each command (or session) in the output. The following is sample output from the **show mgcp commands** command:

```
hostname# show mgcp commands
1 in use, 1 most used, 200 maximum allowed
```

```
CRCX, gateway IP: host-pc-2, transaction ID: 2052, idle: 0:00:07
```

The following is sample output from the **show mgcp detail** command.

```
hostname# show mgcp commands detail
1 in use, 1 most used, 200 maximum allowed
CRCX, idle: 0:00:10
        Gateway IP      host-pc-2
        Transaction ID  2052
        Endpoint name   aaln/1
        Call ID         9876543210abcdef
        Connection ID
        Media IP        192.168.5.7
        Media port      6058
```

The following is sample output from the **show mgcp sessions** command.

```
hostname# show mgcp sessions
1 in use, 1 most used
Gateway IP host-pc-2, connection ID 6789af54c9, active 0:00:11
```

The following is sample output from the **show mgcp sessions detail** command.

```
hostname# show mgcp sessions detail
1 in use, 1 most used
Session active 0:00:14
        Gateway IP      host-pc-2
        Call ID         9876543210abcdef
        Connection ID   6789af54c9
        Endpoint name   aaln/1
        Media lcl port  6166
        Media rmt IP    192.168.5.7
        Media rmt port  6058
```

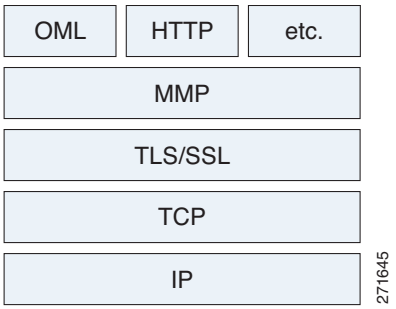
MMP Inspection

The security appliance includes an inspection engine to validate the CUMA Mobile Multiplexing Protocol (MMP).

For information about setting up the TLS Proxy for the Mobility Advantage feature, see Cisco Unified Mobility and MMP Inspection Engine, page 27-52.

MMP is a data transport protocol for transmitting data entities between CUMA clients and servers. As shown in Figure 26-5, MMP must be run on top of a connection-oriented protocol (the underlying transport) and is intended to be run on top of a secure transport protocol such as TLS. The Orative Markup Language (OML) protocol is intended to be run on top of MMP for the purposes of data synchronization, as well as the HTTP protocol for uploading and downloading large files.

Figure 26-5 MMP Stack



The TCP/TLS default port is 5443. There are no embedded NAT or secondary connections.

CUMA client and server communications can be proxied via TLS, which decrypts the data, passes it to the inspect MMP module, and re-encrypt the data before forwarding it to the endpoint. The inspect MMP module verifies the integrity of the MMP headers and passes the OML/HTTP to an appropriate handler. The security appliance takes the following actions on the MMP headers and data:

- Verifies that client MMP headers are well-formed. Upon detection of a malformed header, the TCP session is terminated.
- Verifies that client to server MMP header lengths are not exceeded. If an MMP header length is exceeded (4096), then the TCP session is terminated.
- Verifies that client to server MMP content lengths are not exceeded. If an entity content length is exceeded (4096), the TCP session is terminated.

Note 4096 is the value currently used in MMP implementations.

Since MMP headers and entities can be split across packets, the security appliance buffers data to ensure consistent inspection. The SAPI (stream API) handles data buffering for pending inspection opportunities. MMP header text is treated as case insensitive and a space is present between header text and values. Reclaiming of MMP state is performed by monitoring the state of the TCP connection. Timeouts for these connections follow existing configurable values via the **timeout** command.

MMP inspection is disabled by default. When enabled, MMP inspection operates on TCP destination and source port 5443.

Configuring MMP Inspection for a TLS Proxy

The following procedure provides the steps to configure MMP inspection for a TLS proxy.

However, if you must configure a TLS proxy for the Mobility Advantage feature, see Cisco Unified Mobility and MMP Inspection Engine, page 27-52 for information about all the steps required to make TLS Proxy fully functional.

Step 1 Create the class map by entering the following command:

```
hostname(config)# class-map class_map_name
```

Where *class_map_name* is the name of the class map.

Step 2 Configure the port by entering the following command:

```
hostname(config-cmap)# match port tcp eq port
```

Step 3 Return to global configuration mode by entering the following command:

```
hostname(config-cmap)# exit
```

Step 4 Create the policy map by entering the following command:

```
hostname(config)# policy-map name
```

Use the **policy-map** command (without the **type** keyword) to assign actions to traffic that you identified with a Layer 3/4 class map.

Step 5 Assign a class map to the policy map where you can assign actions to the class map traffic by entering the following command:

```
hostname(config-pmap)# class classmap_name
```

Step 6 Configure the MMP inspection engine by entering the following command:

```
hostname(config-pmap)# inspect mmp tls-proxy name
```

Where *name* specifies the TLS proxy instance name. Entering the **tls-proxy** keyword enables the TLS proxy for MMP inspection. The MMP protocol can additionally use the TCP transport; however, the CUMA client only supports the TLS transport. Therefore, the tls-proxy keyword is required to enable MMP inspection.

Step 7 Return to global configuration mode by entering the following command:

```
hostname(config-pmap)# exit
```

Step 8 Configure the service policy by entering the following command:

```
hostname(config)# service-policy policy_map_name global
```

NetBIOS Inspection

NetBIOS inspection is enabled by default. The NetBios inspection engine translates IP addresses in the NetBios name service (NBNS) packets according to the security appliance NAT configuration.

Configuring a NetBIOS Inspection Policy Map for Additional Inspection Control

To specify actions when a message violates a parameter, create a NETBIOS inspection policy map. You can then apply the inspection policy map when you enable NETBIOS inspection according to the "Configuring Application Inspection" section on page 26-5.

To create a NETBIOS inspection policy map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the "Creating a Regular Expression" section on page 16-13. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the "Creating a Regular Expression Class Map" section on page 16-16.

Step 3 Create a NetBIOS inspection policy map, enter the following command:

```
hostname(config)# policy-map type inspect netbios policy_map_name
hostname(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 4 (Optional) To add a description to the policy map, enter the following command:

```
hostname(config-pmap)# description string
```

Step 5 To apply actions to matching traffic, perform the following steps.

a. Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the NetBIOS class map that you created in Step 3 by entering the following command:

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

- Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
hostname(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

Not all options are available for each **match** or **class** command. See the CLI help or the *Cisco Security Appliance Command Reference* for the exact options available.

The **drop** keyword drops all packets that match.

The **send-protocol-error** keyword sends a protocol error message.

The **drop-connection** keyword drops the packet and closes the connection.

The **mask** keyword masks out the matching portion of the packet.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

The **rate-limit** *message_rate* argument limits the rate of messages.

You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 16-9.

Step 6 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

b. To check for NETBIOS protocol violations, enter the following command:

```
hostname(config-pmap-p)# protocol-violation [action [drop-connection | reset | log]]
```

Where the **drop-connection** action closes the connection. The **reset** action closes the connection and sends a TCP reset to the client. The **log** action sends a system log message when this policy map matches traffic.

The following example shows how to define a NETBIOS inspection policy map.

```
hostname(config)# policy-map type inspect netbios netbios_map
hostname(config-pmap)# protocol-violation drop log

hostname(config)# policy-map netbios_policy
hostname(config-pmap)# class inspection_default
hostname(config-pmap-c)# inspect netbios netbios_map
```

PPTP Inspection

PPTP is a protocol for tunneling PPP traffic. A PPTP session is composed of one TCP channel and usually two PPTP GRE tunnels. The TCP channel is the control channel used for negotiating and managing the PPTP GRE tunnels. The GRE tunnels carries PPP sessions between the two hosts.

When enabled, PPTP application inspection inspects PPTP protocol packets and dynamically creates the GRE connections and xlates necessary to permit PPTP traffic. Only Version 1, as defined in RFC 2637, is supported.

PAT is only performed for the modified version of GRE [RFC 2637] when negotiated over the PPTP TCP control channel. Port Address Translation is *not* performed for the unmodified version of GRE [RFC 1701, RFC 1702].

Specifically, the security appliance inspects the PPTP version announcements and the outgoing call request/response sequence. Only PPTP Version 1, as defined in RFC 2637, is inspected. Further inspection on the TCP control channel is disabled if the version announced by either side is not Version 1. In addition, the outgoing-call request and reply sequence are tracked. Connections and xlates are dynamic allocated as necessary to permit subsequent secondary GRE data traffic.

The PPTP inspection engine must be enabled for PPTP traffic to be translated by PAT. Additionally, PAT is only performed for a modified version of GRE (RFC2637) and only if it is negotiated over the PPTP TCP control channel. PAT is not performed for the unmodified version of GRE (RFC 1701 and RFC 1702).

As described in RFC 2637, the PPTP protocol is mainly used for the tunneling of PPP sessions initiated from a modem bank PAC (PPTP Access Concentrator) to the headend PNS (PPTP Network Server). When used this way, the PAC is the remote client and the PNS is the server.

However, when used for VPN by Windows, the interaction is inverted. The PNS is a remote single-user PC that initiates connection to the head-end PAC to gain access to a central network.

RADIUS Accounting Inspection

One of the well known problems is the over-billing attack in GPRS networks. The over-billing attack can cause consumers anger and frustration by being billed for services that they have not used. In this case, a malicious attacker sets up a connection to a server and obtains an IP address from the SGSN. When the attacker ends the call, the malicious server will still send packets to it, which gets dropped by the GGSN, but the connection from the server remains active. The IP address assigned to the malicious attacker gets released and reassigned to a legitimate user who will then get billed for services that the attacker will use.

RADIUS accounting inspection prevents this type of attack by ensuring the traffic seen by the GGSN is legitimate. With the RADIUS accounting feature properly configured, the security appliance tears down a connection based on matching the Framed IP attribute in the Radius Accounting Request Start message

with the Radius Accounting Request Stop message. When the Stop message is seen with the matching IP address in the Framed IP attribute, the security appliance looks for all connections with the source matching the IP address.

You have the option to configure a secret pre-shared key with the RADIUS server so the security appliance can validate the message. If the shared secret is not configured, the security appliance does not need to validate the source of the message and will only check that the source IP address is one of the configured addresses allowed to send the RADIUS messages.

Note When using RADIUS accounting inspection with GPRS enabled, thesecurity appliance checks for the 3GPP-Session-Stop-Indicator in the Accounting Request STOP messages to properly handle secondary PDP contexts. Specifically, the security appliance requires that the Accounting Request STOP messages include the 3GPP-SGSN-Address attribute before it will temrinate the user sessions and all associated connections. Some third-party GGSNs might not send this attribute by default.

Configuring a RADIUS Inspection Policy Map for Additional Inspection Control

In order to use this feature, the **radius-accounting-map** will need to be specified in the **policy-map** and then applied to the service-policy to specify that this traffic is for to-the-box inspection.

The following example shows the complete set of commands in context to properly configure this feature:

Step 1 Configure the class map and the port:

```
class-map type management c1
  match port udp eq 1813
```

Step 2 Create the policy map, and configure the parameters for RADIUS accounting inspection using the parameter command to access the proper mode to configure the attributes, host, and key.

```
policy-map type inspect radius-accounting radius_accounting_map
  parameters
    host 10.1.1.1 inside key 123456789
    send response
    enable gprs
    validate-attribute 31
```

Step 3 Configure the service policy.

```
policy-map global_policy
  class c1
     inspect radius-accounting radius_accounting_map

service-policy global_policy global
```

RSH Inspection

RSH inspection is enabled by default. The RSH protocol uses a TCP connection from the RSH client to the RSH server on TCP port 514. The client and server negotiate the TCP port number where the client listens for the STDERR output stream. RSH inspection supports NAT of the negotiated port number if necessary.

RTSP Inspection

This section describes RTSP application inspection. This section includes the following topics:

- RTSP Inspection Overview, page 26-65
- Using RealPlayer, page 26-65
- Restrictions and Limitations, page 26-66

RTSP Inspection Overview

The RTSP inspection engine lets the security appliance pass RTSP packets. RTSP is used by RealAudio, RealNetworks, Apple QuickTime 4, RealPlayer, and Cisco IP/TV connections.

Note For Cisco IP/TV, use RTSP TCP port 554 and TCP 8554.

RTSP applications use the well-known port 554 with TCP (rarely UDP) as a control channel. The security appliance only supports TCP, in conformity with RFC 2326. This TCP control channel is used to negotiate the data channels that is used to transmit audio/video traffic, depending on the transport mode that is configured on the client.

The supported RDT transports are: rtp/avp, rtp/avp/udp, x-real-rdt, x-real-rdt/udp, and x-pn-tng/udp.

The security appliance parses Setup response messages with a status code of 200. If the response message is travelling inbound, the server is outside relative to the security appliance and dynamic channels need to be opened for connections coming inbound from the server. If the response message is outbound, then the security appliance does not need to open dynamic channels.

Because RFC 2326 does not require that the client and server ports must be in the SETUP response message, the security appliance keeps state and remembers the client ports in the SETUP message. QuickTime places the client ports in the SETUP message and then the server responds with only the server ports.

RTSP inspection does not support PAT or dual-NAT. Also, the security appliance cannot recognize HTTP cloaking where RTSP messages are hidden in the HTTP messages.

Using RealPlayer

When using RealPlayer, it is important to properly configure transport mode. For the security appliance, add an **access-list** command from the server to the client or vice versa. For RealPlayer, change transport mode by clicking **Options>Preferences>Transport>RTSP Settings**.

If using TCP mode on the RealPlayer, select the **Use TCP to Connect to Server** and **Attempt to use TCP for all content** check boxes. On the security appliance, there is no need to configure the inspection engine.

If using UDP mode on the RealPlayer, select the **Use TCP to Connect to Server** and **Attempt to use UDP for static content** check boxes, and for live content not available via Multicast. On the security appliance, add an **inspect rtsp** *port* command.

As a call is set up, the SIP session is in the "transient" state until the media address and media port is received from the called endpoint in a Response message indicating the RTP port the called endpoint listens on. If there is a failure to receive the response messages within one minute, the signaling connection is torn down.

Once the final handshake is made, the call state is moved to active and the signaling connection remains until a BYE message is received.

If an inside endpoint initiates a call to an outside endpoint, a media hole is opened to the outside interface to allow RTP/RTCP UDP packets to flow to the inside endpoint media address and media port specified in the INVITE message from the inside endpoint. Unsolicited RTP/RTCP UDP packets to an inside interface does not traverse the security appliance, unless the security appliance configuration specifically allows it.

Configuring a SIP Inspection Policy Map for Additional Inspection Control

To specify actions when a message violates a parameter, create a SIP inspection policy map. You can then apply the inspection policy map when you enable SIP inspection according to the "Configuring Application Inspection" section on page 26-5.

To create a SIP inspection policy map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the "Creating a Regular Expression" section on page 16-13. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the "Creating a Regular Expression Class Map" section on page 16-16.s

Step 3 (Optional) Create a SIP inspection class map by performing the following steps.

A class map groups multiple traffic matches. Traffic must match *all* of the **match** commands to match the class map. You can alternatively identify **match** commands directly in the policy map. The difference between creating a class map and defining the traffic match directly in the inspection policy map is that the class map lets you create more complex match criteria, and you can reuse class maps.

To specify traffic that should not match the class map, use the **match not** command. For example, if the **match not** command specifies the string "example.com," then any traffic that includes "example.com" does not match the class map.

For the traffic that you identify in this class map, you can specify actions such as drop-connection, reset, and/or log the connection in the inspection policy map.

If you want to perform different actions for each **match** command, you should identify the traffic directly in the policy map.

a. Create the class map by entering the following command:

```
hostname(config)# class-map type inspect sip [match-all | match-any] class_map_name
hostname(config-cmap)#
```

Where *the class_map_name* is the name of the class map. The **match-all** keyword is the default, and specifies that traffic must match all criteria to match the class map. The **match-any** keyword specifies that the traffic matches the class map if it matches at least one of the criteria. The CLI enters class-map configuration mode, where you can enter one or more **match** commands.

b. (Optional) To add a description to the class map, enter the following command:

```
hostname(config-cmap)# description string
```

Where *class_map_name* is the name of the class map. The **match-all** keyword is the default, and specifies that traffic must match all criteria to match the class map. The **match-any** keyword specifies that the traffic matches the class map if it matches at least one of the criteria. The CLI enters class-map configuration mode, where you can enter one or more **match** commands.

b. (Optional) To add a description to the class map, enter the following command:

```
hostname(config-cmap)# description string
```

c. (Optional) To match an RTSP request method, enter the following command:

```
hostname(config-cmap)# match [not] request-method method
```

Where *method* is the type of method to match (announce, describe, get_parameter, options, pause, play, record, redirect, setup, set_parameter, teardown).

d. (Optional) To match URL filtering, enter the following command:

```
hostname(config-cmap)# match [not] url-filter regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

Step 4 To create an RTSP inspection policy map, enter the following command:

```
hostname(config)# policy-map type inspect rtsp policy_map_name
hostname(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 5 (Optional) To add a description to the policy map, enter the following command:

```
hostname(config-pmap)# description string
```

Step 6 To apply actions to matching traffic, perform the following steps.

a. Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the RTSP class map that you created in Step 3 by entering the following command:

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

- Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
hostname(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

Not all options are available for each **match** or **class** command. See the CLI help or the *Cisco Security Appliance Command Reference* for the exact options available.

The **drop** keyword drops all packets that match.

The **send-protocol-error** keyword sends a protocol error message.

The **drop-connection** keyword drops the packet and closes the connection.

The **mask** keyword masks out the matching portion of the packet.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

The **rate-limit** *message_rate* argument limits the rate of messages.

You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 16-9.

Step 7 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

b. To restrict usage on reserve port for media negotiation, enter the following command:

```
hostname(config-pmap-p)# reserve-port-protect
```

c. To set the limit on the URL length allowed in the message, enter the following command:

```
hostname(config-pmap-p)# url-length-limit length
```

Where the *length* argument specifies the URL length in bytes (0 to 6000).

The following example shows a how to define an RTSP inspection policy map.

```
hostname(config)# regex badurl1 www.url1.com/rtsp.avi
hostname(config)# regex badurl2 www.url2.com/rtsp.rm
hostname(config)# regex badurl3 www.url3.com/rtsp.asp

hostname(config)# class-map type regex match-any badurl-list
hostname(config-cmap)# match regex badurl1
hostname(config-cmap)# match regex badurl2
hostname(config-cmap)# match regex badurl3

hostname(config)# policy-map type inspect rtsp rtsp-filter-map
hostname(config-pmap)# match url-filter regex class badurl-list
hostname(config-pmap-p)# drop-connection

hostname(config)# class-map rtsp-traffic-class
hostname(config-cmap)# match default-inspection-traffic

hostname(config)# policy-map rtsp-traffic-policy
hostname(config-pmap)# class rtsp-traffic-class
hostname(config-pmap-c)# inspect rtsp rtsp-filter-map

hostname(config)# service-policy rtsp-traffic-policy global
```

SIP Inspection

This section describes SIP application inspection. This section includes the following topics:

- SIP Inspection Overview, page 26-69
- SIP Instant Messaging, page 26-69
- Configuring SIP Timeout Values, page 26-74
- Verifying and Monitoring SIP Inspection, page 26-74

SIP Inspection Overview

SIP, as defined by the IETF, enables call handling sessions, particularly two-party audio conferences, or "calls." SIP works with SDP for call signalling. SDP specifies the ports for the media stream. Using SIP, the security appliance can support any SIP VoIP gateways and VoIP proxy servers. SIP and SDP are defined in the following RFCs:

- SIP: Session Initiation Protocol, RFC 3261
- SDP: Session Description Protocol, RFC 2327

To support SIP calls through the security appliance, signaling messages for the media connection addresses, media ports, and embryonic connections for the media must be inspected, because while the signaling is sent over a well-known destination port (UDP/TCP 5060), the media streams are dynamically allocated. Also, SIP embeds IP addresses in the user-data portion of the IP packet. SIP inspection applies NAT for these embedded IP addresses.

The following limitations and restrictions apply when using PAT with SIP:

- If a remote endpoint tries to register with a SIP proxy on a network protected by the security appliance, the registration fails under very specific conditions, as follows:
  - PAT is configured for the remote endpoint.
  - The SIP registrar server is on the outside network.
  - The port is missing in the contact field in the REGISTER message sent by the endpoint to the proxy server.
- If a SIP device transmits a packet in which the SDP portion has an IP address in the owner/creator field (o=) that is different than the IP address in the connection field (c=), the IP address in the o= field may not be properly translated. This is due to a limitation in the SIP protocol, which does not provide a port value in the o= field.

The following limitation applies when performing SIP inspection on the PIX 500 Series platform.

The PIX 500 Series platform does not support transmitting and inspecting SIP messages over 6144 bytes because the 8192-byte reassembly limit (the TCP proxy limit) is larger than the security appliance 6144 byte message limit. This limitation can be encountered when Cisco Unified Personal Communicator clients download large buddy lists and the SIP traffic must be inspected by the security appliance. This limitation does not exist on the ASA 5500 Series platform.

SIP Instant Messaging

Instant Messaging refers to the transfer of messages between users in near real-time. SIP supports the Chat feature on Windows XP using Windows Messenger RTC Client version 4.7.0105 only. The MESSAGE/INFO methods and 202 Accept response are used to support IM as defined in the following RFCs:

- Session Initiation Protocol (SIP)-Specific Event Notification, RFC 3265
- Session Initiation Protocol (SIP) Extension for Instant Messaging, RFC 3428

MESSAGE/INFO requests can come in at any time after registration/subscription. For example, two users can be online at any time, but not chat for hours. Therefore, the SIP inspection engine opens pinholes that time out according to the configured SIP timeout value. This value must be configured at least five minutes longer than the subscription duration. The subscription duration is defined in the Contact Expires value and is typically 30 minutes.

Because MESSAGE/INFO requests are typically sent using a dynamically allocated port other than port 5060, they are required to go through the SIP inspection engine.

Note Only the Chat feature is currently supported. Whiteboard, File Transfer, and Application Sharing are not supported. RTC Client 5.0 is not supported.

SIP inspection translates the SIP text-based messages, recalculates the content length for the SDP portion of the message, and recalculates the packet length and checksum. It dynamically opens media connections for ports specified in the SDP portion of the SIP message as address/ports on which the endpoint should listen.

SIP inspection has a database with indices CALL_ID/FROM/TO from the SIP payload. These indices identify the call, the source, and the destination. This database contains the media addresses and media ports found in the SDP media information fields and the media type. There can be multiple media addresses and ports for a session. The security appliance opens RTP/RTCP connections between the two endpoints using these media addresses/ports.

The well-known port 5060 must be used on the initial call setup (INVITE) message; however, subsequent messages may not have this port number. The SIP inspection engine opens signaling connection pinholes, and marks these connections as SIP connections. This is done for the messages to reach the SIP application and be translated.

As a call is set up, the SIP session is in the “transient” state until the media address and media port is received from the called endpoint in a Response message indicating the RTP port the called endpoint listens on. If there is a failure to receive the response messages within one minute, the signaling connection is torn down.

Once the final handshake is made, the call state is moved to active and the signaling connection remains until a BYE message is received.

If an inside endpoint initiates a call to an outside endpoint, a media hole is opened to the outside interface to allow RTP/RTCP UDP packets to flow to the inside endpoint media address and media port specified in the INVITE message from the inside endpoint. Unsolicited RTP/RTCP UDP packets to an inside interface does not traverse the security appliance, unless the security appliance configuration specifically allows it.

Configuring a SIP Inspection Policy Map for Additional Inspection Control

To specify actions when a message violates a parameter, create a SIP inspection policy map. You can then apply the inspection policy map when you enable SIP inspection according to the “Configuring Application Inspection” section on page 26-5.

To create a SIP inspection policy map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the “Creating a Regular Expression” section on page 16-13. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the “Creating a Regular Expression Class Map” section on page 16-16.s

Step 3 (Optional) Create a SIP inspection class map by performing the following steps.

A class map groups multiple traffic matches. Traffic must match *all* of the **match** commands to match the class map. You can alternatively identify **match** commands directly in the policy map. The difference between creating a class map and defining the traffic match directly in the inspection policy map is that the class map lets you create more complex match criteria, and you can reuse class maps.

To specify traffic that should not match the class map, use the **match not** command. For example, if the **match not** command specifies the string "example.com," then any traffic that includes "example.com" does not match the class map.

For the traffic that you identify in this class map, you can specify actions such as drop-connection, reset, and/or log the connection in the inspection policy map.

If you want to perform different actions for each **match** command, you should identify the traffic directly in the policy map.

a. Create the class map by entering the following command:

```
hostname(config)# class-map type inspect sip [match-all | match-any] class_map_name
hostname(config-cmap)#
```

Where *the class_map_name* is the name of the class map. The match-all keyword is the default, and specifies that traffic must match all criteria to match the class map. The match-any keyword specifies that the traffic matches the class map if it matches at leX(The CLI enters class-map configuration mode, where you can enter one or more **match** commands.

b. (Optional) To add a description to the class map, enter the following command:

```
hostname(config-cmap)# description string
```

Where *string* is the description of the class map (up to 200 characters).

c. (Optional) To match a called party, as specified in the To header, enter the following command:

```
hostname(config-cmap)# match [not] called-party regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

d. (Optional) To match a calling party, as specified in the From header, enter the following command:

```
hostname(config-cmap)# match [not] calling-party regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

e. (Optional) To match a content length in the SIP header, enter the following command:

```
hostname(config-cmap)# match [not] content length gt length
```

Where *length* is the number of bytes the content length is greater than. 0 to 65536.

f. (Optional) To match an SDP content type or regular expression, enter the following command:

```
hostname(config-cmap)# match [not] content type {sdp | regex {class class_name |
regex_name}}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

g. (Optional) To match a SIP IM subscriber, enter the following command:

```
hostname(config-cmap)# match [not] im-subscriber regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

h. (Optional) To match a SIP via header, enter the following command:

```
hostname(config-cmap)# match [not] message-path regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

i. (Optional) To match a SIP request method, enter the following command:

```
hostname(config-cmap)# match [not] request-method method
```

Where *method* is the type of method to match (ack, bye, cancel, info, invite, message, notify, options, prack, refer, register, subscribe, unknown, update).

j. (Optional) To match the requester of a third-party registration, enter the following command:

```
hostname(config-cmap)# match [not] third-party-registration regex {class class_name |
regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

k. (Optional) To match an URI in the SIP headers, enter the following command:

```
hostname(config-cmap)# match [not] uri {sip | tel} length gt length
```

Where *length* is the number of bytes the URI is greater than. 0 to 65536.

Step 4 Create a SIP inspection policy map, enter the following command:

```
hostname(config)# policy-map type inspect sip policy_map_name
hostname(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 5 (Optional) To add a description to the policy map, enter the following command:

```
hostname(config-pmap)# description string
```

Step 6 To apply actions to matching traffic, perform the following steps.

a. Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the SIP class map that you created in Step 3 by entering the following command:

  ```
  hostname(config-pmap)# class class_map_name
  hostname(config-pmap-c)#
  ```

- Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
hostname(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

Not all options are available for each **match** or **class** command. See the CLI help or the *Cisco Security Appliance Command Reference* for the exact options available.

The **drop** keyword drops all packets that match.

The **send-protocol-error** keyword sends a protocol error message.

The **drop-connection** keyword drops the packet and closes the connection.

The **mask** keyword masks out the matching portion of the packet.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

The **rate-limit** *message_rate* argument limits the rate of messages.

You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 16-9.

Step 7 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

b. To enable or disable instant messaging, enter the following command:

```
hostname(config-pmap-p)# im
```

c. To enable or disable IP address privacy, enter the following command:

```
hostname(config-pmap-p)# ip-address-privacy
```

d. To enable check on Max-forwards header field being 0 (which cannot be 0 before reaching the destination), enter the following command:

```
hostname(config-pmap-p)# max-forwards-validation action {drop | drop-connection |
reset | log} [log]
```

e. To enable check on RTP packets flowing on the pinholes for protocol conformance, enter the following command:

```
hostname(config-pmap-p)# rtp-conformance [enforce-payloadtype]
```

Where the **enforce-payloadtype** keyword enforces the payload type to be audio or video based on the signaling exchange.

f. To identify the Server and User-Agent header fields, which expose the software version of either a server or an endpoint, enter the following command:

```
hostname(config-pmap-p)# software-version action {mask | log} [log]
```

Where the **mask** keyword masks the software version in the SIP messages.

g. To enable state checking validation, enter the following command:

```
hostname(config-pmap-p)# state-checking action {drop | drop-connection | reset | log}
[log]
```

h. To enable strict verification of the header fields in the SIP messages according to RFC 3261, enter the following command:

```
hostname(config-pmap-p)# strict-header-validation action {drop | drop-connection |
reset | log} [log]
```

i. To allow non SIP traffic using the well-known SIP signaling port, enter the following command:

```
hostname(config-pmap-p)# traffic-non-sip
```

j. To identify the non-SIP URIs present in the Alert-Info and Call-Info header fields, enter the following command:

```
hostname(config-pmap-p)# uri-non-sip action {mask | log} [log]
```

The following example shows how to disable instant messaging over SIP:

```
hostname(config)# policy-map type inspect sip mymap
hostname(config-pmap)# parameters
hostname(config-pmap-p)# no im

hostname(config)# policy-map global_policy
hostname(config-pmap)# class inspection_default
hostname(config-pmap-c)# inspect sip mymap

hostname(config)# service-policy global_policy global
```

Configuring SIP Timeout Values

The media connections are torn down within two minutes after the connection becomes idle. This is, however, a configurable timeout and can be set for a shorter or longer period of time. To configure the timeout for the SIP control connection, enter the following command:

```
hostname(config)# timeout sip hh:mm:ss
```

This command configures the idle timeout after which a SIP control connection is closed.

To configure the timeout for the SIP media connection, enter the following command:

```
hostname(config)# timeout sip_media hh:mm:ss
```

This command configures the idle timeout after which a SIP media connection is closed.

Verifying and Monitoring SIP Inspection

The **show sip** command assists in troubleshooting SIP inspection engine issues and is described with the **inspect protocol sip udp 5060** command. The **show timeout sip** command displays the timeout value of the designated protocol.

The **show sip** command displays information for SIP sessions established across the security appliance. Along with the **debug sip** and **show local-host** commands, this command is used for troubleshooting SIP inspection engine issues.

Note We recommend that you configure the **pager** command before entering the **show sip** command. If there are a lot of SIP session records and the **pager** command is not configured, it takes a while for the **show sip** command output to reach its end.

The following is sample output from the **show sip** command:

```
hostname# show sip
Total: 2
call-id c3943000-960ca-2e43-228f@10.130.56.44
    state Call init, idle 0:00:01
call-id c3943000-860ca-7e1f-11f7@10.130.56.45
    state Active, idle 0:00:06
```

This sample shows two active SIP sessions on the security appliance (as shown in the Total field). Each call-id represents a call.

The first session, with the call-id c3943000-960ca-2e43-228f@10.130.56.44, is in the state Call Init, which means the session is still in call setup. Call setup is not complete until a final response to the call has been received. For instance, the caller has already sent the INVITE, and maybe received a 100 Response, but has not yet seen the 200 OK, so the call setup is not complete yet. Any non-1xx response message is considered a final response. This session has been idle for 1 second.

The second session is in the state Active, in which call setup is complete and the endpoints are exchanging media. This session has been idle for 6 seconds.

Skinny (SCCP) Inspection

This section describes SCCP application inspection. This section includes the following topics:

- SCCP Inspection Overview, page 26-75
- Supporting Cisco IP Phones, page 26-75
- Restrictions and Limitations, page 26-76
- Verifying and Monitoring SCCP Inspection, page 26-76

SCCP Inspection Overview

Skinny (SCCP) is a simplified protocol used in VoIP networks. Cisco IP Phones using SCCP can coexist in an H.323 environment. When used with Cisco CallManager, the SCCP client can interoperate with H.323 compliant terminals. Application layer functions in the security appliance recognize SCCP Version 3.3. There are 5 versions of the SCCP protocol: 2.4, 3.0.4, 3.1.1, 3.2, and 3.3.2. The security appliance supports all versions through Version 3.3.2.

The security appliance supports PAT and NAT for SCCP. PAT is necessary if you have more IP phones than global IP addresses for the IP phones to use. By supporting NAT and PAT of SCCP Signaling packets, Skinny application inspection ensures that all SCCP signalling and media packets can traverse the security appliance.

Normal traffic between Cisco CallManager and Cisco IP Phones uses SCCP and is handled by SCCP inspection without any special configuration. The security appliance also supports DHCP options 150 and 66, which it accomplishes by sending the location of a TFTP server to Cisco IP Phones and other DHCP clients. Cisco IP Phones might also include DHCP option 3 in their requests, which sets the default route. For more information, see the "Using Cisco IP Phones with a DHCP Server" section on page 11-4.

Supporting Cisco IP Phones

In topologies where Cisco CallManager is located on the higher security interface with respect to the Cisco IP Phones, if NAT is required for the Cisco CallManager IP address, the mapping must be **static** as a Cisco IP Phone requires the Cisco CallManager IP address to be specified explicitly in its configuration. An static identity entry allows the Cisco CallManager on the higher security interface to accept registrations from the Cisco IP Phones.

Cisco IP Phones require access to a TFTP server to download the configuration information they need to connect to the Cisco CallManager server.

When the Cisco IP Phones are on a lower security interface compared to the TFTP server, you must use an access list to connect to the protected TFTP server on UDP port 69. While you do need a static entry for the TFTP server, this does not have to be an identity static entry. When using NAT, an identity static entry maps to the same IP address. When using PAT, it maps to the same IP address and port.

When the Cisco IP Phones are on a *higher* security interface compared to the TFTP server and Cisco CallManager, no access list or static entry is required to allow the Cisco IP Phones to initiate the connection.

Restrictions and Limitations

The following are limitations that apply to the current version of PAT and NAT support for SCCP:

- PAT does not work with configurations containing the **alias** command.
- Outside NAT or PAT is *not* supported.

If the address of an internal Cisco CallManager is configured for NAT or PAT to a different IP address or port, registrations for external Cisco IP Phones fail because the security appliance currently does not support NAT or PAT for the file content transferred over TFTP. Although the security appliance supports NAT of TFTP messages and opens a pinhole for the TFTP file, the security appliance cannot translate the Cisco CallManager IP address and port embedded in the Cisco IP Phone configuration files that are transferred by TFTP during phone registration.



Note The security appliance supports stateful failover of SCCP calls except for calls that are in the middle of call setup.

Verifying and Monitoring SCCP Inspection

The **show skinny** command assists in troubleshooting SCCP (Skinny) inspection engine issues. The following is sample output from the **show skinny** command under the following conditions. There are two active Skinny sessions set up across the security appliance. The first one is established between an internal Cisco IP Phone at local address 10.0.0.11 and an external Cisco CallManager at 172.18.1.33. TCP port 2000 is the CallManager. The second one is established between another internal Cisco IP Phone at local address 10.0.0.22 and the same Cisco CallManager.

```
hostname# show skinny
        LOCAL                   FOREIGN                 STATE
----------------------------------------------------------------
1       10.0.0.11/52238         172.18.1.33/2000                1
  MEDIA 10.0.0.11/22948         172.18.1.22/20798
2       10.0.0.22/52232         172.18.1.33/2000                1
  MEDIA 10.0.0.22/20798         172.18.1.11/22948
```

The output indicates that a call has been established between two internal Cisco IP Phones. The RTP listening ports of the first and second phones are UDP 22948 and 20798 respectively.

The following is sample output from the **show xlate debug** command for these Skinny connections:

```
hostname# show xlate debug
2 in use, 2 most used
Flags:  D - DNS, d - dump, I - identity, i - inside, n - no random,
        r - portmap, s - static
NAT from inside:10.0.0.11 to outside:172.18.1.11 flags si idle 0:00:16 timeout 0:05:00
NAT from inside:10.0.0.22 to outside:172.18.1.22 flags si idle 0:00:14 timeout 0:05:00
```

Configuring a Skinny (SCCP) Inspection Policy Map for Additional Inspection Control

To specify actions when a message violates a parameter, create an SCCP inspection policy map. You can then apply the inspection policy map when you enable SCCP inspection according to the "Configuring Application Inspection" section on page 26-5.

To create an SCCP inspection policy map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the "Creating a Regular Expression" section on page 16-13. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the "Creating a Regular Expression Class Map" section on page 16-16.

Step 3 Create an SCCP inspection policy map, enter the following command:

```
hostname(config)# policy-map type inspect skinny policy_map_name
hostname(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 4 (Optional) To add a description to the policy map, enter the following command:

```
hostname(config-pmap)# description string
```

Step 5 To apply actions to matching traffic, perform the following steps.

a. Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the SCCP class map that you created in Step 3 by entering the following command:

  ```
  hostname(config-pmap)# class class_map_name
  hostname(config-pmap-c)#
  ```

- Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
hostname(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

Not all options are available for each **match** or **class** command. See the CLI help or the *Cisco Security Appliance Command Reference* for the exact options available.

The **drop** keyword drops all packets that match.

The **send-protocol-error** keyword sends a protocol error message.

The **drop-connection** keyword drops the packet and closes the connection.

The **mask** keyword masks out the matching portion of the packet.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

The **rate-limit** *message_rate* argument limits the rate of messages.

Step 6 You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 16-9.To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

b. To enforce registration before calls can be placed, enter the following command:

```
hostname(config-pmap-p)# enforce-registration
```

c. To set the maximum SCCP station message ID allowed, enter the following command:

```
hostname(config-pmap-p)# message-ID max hex_value
```

Where the *hex_value* argument is the station message ID in hex.

d. To check RTP packets flowing on the pinholes for protocol conformance, enter the following command:

```
hostname(config-pmap-p)# rtp-conformance [enforce-payloadtype]
```

Where the **enforce-payloadtype** keyword enforces the payload type to be audio or video based on the signaling exchange.

e. To set the maximum and minimum SCCP prefix length value allowed, enter the following command:

```
hostname(config-pmap-p)# sccp-prefix-len {max | min} value_length
```

Where the *value_length* argument is a maximum or minimum value.

f. To configure the timeout value for signaling and media connections, enter the following command:

```
hostname(config-pmap-p)# timeout
```

The following example shows how to define an SCCP inspection policy map.

```
hostname(config)# policy-map type inspect skinny skinny-map
hostname(config-pmap)# parameters
hostname(config-pmap-p)# enforce-registration
hostname(config-pmap-p)# match message-id range 200 300
hostname(config-pmap-p)# drop log
hostname(config)# class-map inspection_default
hostname(config-cmap)# match default-inspection-traffic
hostname(config)# policy-map global_policy
hostname(config-pmap)# class inspection_default
hostname(config-pmap-c)# inspect skinny skinny-map
hostname(config)# service-policy global_policy global
```

SMTP and Extended SMTP Inspection

ESMTP application inspection provides improved protection against SMTP-based attacks by restricting the types of SMTP commands that can pass through the security appliance and by adding monitoring capabilities.

ESMTP is an enhancement to the SMTP protocol and is similar is most respects to SMTP. For convenience, the term SMTP is used in this document to refer to both SMTP and ESMTP. The application inspection process for extended SMTP is similar to SMTP application inspection and includes support for SMTP sessions. Most commands used in an extended SMTP session are the same as those used in an SMTP session but an ESMTP session is considerably faster and offers more options related to reliability and security, such as delivery status notification.

Extended SMTP application inspection adds support for eight extended SMTP commands, including AUTH, EHLO, ETRN, HELP, SAML, SEND, SOML and VRFY. Along with the support for seven RFC 821 commands (DATA, HELO, MAIL, NOOP, QUIT, RCPT, RSET), the security appliance supports a total of fifteen SMTP commands.

Other extended SMTP commands, such as ATRN, STARTLS, ONEX, VERB, CHUNKING, and private extensions and are not supported. Unsupported commands are translated into Xs, which are rejected by the internal server. This results in a message such as “500 Command unknown: 'XXX'.” Incomplete commands are discarded.

The ESMTP inspection engine changes the characters in the server SMTP banner to asterisks except for the “2”, “0”, “0” characters. Carriage return (CR) and linefeed (LF) characters are ignored.

With SMTP inspection enabled, a Telnet session used for interactive SMTP may hang if the following rules are not observed: SMTP commands must be at least four characters in length; must be terminated with carriage return and line feed; and must wait for a response before issuing the next reply.

An SMTP server responds to client requests with numeric reply codes and optional human-readable strings. SMTP application inspection controls and reduces the commands that the user can use as well as the messages that the server returns. SMTP inspection performs three primary tasks:

- Restricts SMTP requests to seven basic SMTP commands and eight extended commands.
- Monitors the SMTP command-response sequence.
- Generates an audit trail—Audit record 108002 is generated when invalid character embedded in the mail address is replaced. For more information, see RFC 821.

SMTP inspection monitors the command and response sequence for the following anomalous signatures:

- Truncated commands.
- Incorrect command termination (not terminated with <CR><LR>).
- The MAIL and RCPT commands specify who are the sender and the receiver of the mail. Mail addresses are scanned for strange characters. The pipeline character (|) is deleted (changed to a blank space) and “<” ,”>” are only allowed if they are used to define a mail address (“>” must be preceded by “<”).
- Unexpected transition by the SMTP server.
- For unknown commands, the security appliance changes all the characters in the packet to X. In this case, the server generates an error code to the client. Because of the change in the packed, the TCP checksum has to be recalculated or adjusted.
- TCP stream editing.
- Command pipelining.

SNMP Inspection

SNMP application inspection lets you restrict SNMP traffic to a specific version of SNMP. Earlier versions of SNMP are less secure; therefore, denying certain SNMP versions may be required by your security policy. The security appliance can deny SNMP versions 1, 2, 2c, or 3. You control the versions permitted by creating an SNMP map. You then apply the SNMP map when you enable SNMP inspection according to the "Configuring Application Inspection" section on page 26-5.

To create an SNMP inspection policy map, perform the following steps:

Step 1 To create an SNMP map, enter the following command:

```
hostname(config)# snmp-map map_name
hostname(config-snmp-map)#
```

where *map_name* is the name of the SNMP map. The CLI enters SNMP map configuration mode.

Step 2 To specify the versions of SNMP to deny, enter the following command for each version:

```
hostname(config-snmp-map)# deny version version
hostname(config-snmp-map)#
```

where *version* is 1, 2, 2c, or 3.

The following example denies SNMP Versions 1 and 2:

```
hostname(config)# snmp-map sample_map
hostname(config-snmp-map)# deny version 1
hostname(config-snmp-map)# deny version 2
```

SQL*Net Inspection

SQL*Net inspection is enabled by default.

The SQL*Net protocol consists of different packet types that the security appliance handles to make the data stream appear consistent to the Oracle applications on either side of the security appliance.

The default port assignment for SQL*Net is 1521. This is the value used by Oracle for SQL*Net, but this value does not agree with IANA port assignments for Structured Query Language (SQL). Use the **class-map** command to apply SQL*Net inspection to a range of port numbers.

Note Disable SQL*Net inspection when SQL data transfer occurs on the same port as the SQL control TCP port 1521. The security appliance acts as a proxy when SQL*Net inspection is enabled and reduces the client window size from 65000 to about 16000 causing data transfer issues.

The security appliance translates all addresses and looks in the packets for all embedded ports to open for SQL*Net Version 1.

For SQL*Net Version 2, all DATA or REDIRECT packets that immediately follow REDIRECT packets with a zero data length will be fixed up.

The packets that need fix-up contain embedded host/port addresses in the following format:

```
(ADDRESS=(PROTOCOL=tcp)(DEV=6)(HOST=a.b.c.d)(PORT=a))
```

SQL*Net Version 2 TNSFrame types (Connect, Accept, Refuse, Resend, and Marker) will not be scanned for addresses to NAT nor will inspection open dynamic connections for any embedded ports in the packet.

SQL*Net Version 2 TNSFrames, Redirect, and Data packets will be scanned for ports to open and addresses to NAT, if preceded by a REDIRECT TNSFrame type with a zero data length for the payload. When the Redirect message with data length zero passes through the security appliance, a flag will be set in the connection data structure to expect the Data or Redirect message that follows to be translated and ports to be dynamically opened. If one of the TNS frames in the preceding paragraph arrive after the Redirect message, the flag will be reset.

The SQL*Net inspection engine will recalculate the checksum, change IP, TCP lengths, and readjust Sequence Numbers and Acknowledgment Numbers using the delta of the length of the new and old message.

SQL*Net Version 1 is assumed for all other cases. TNSFrame types (Connect, Accept, Refuse, Resend, Marker, Redirect, and Data) and all packets will be scanned for ports and addresses. Addresses will be translated and port connections will be opened.

Sun RPC Inspection

This section describes Sun RPC application inspection. This section includes the following topics:

- Sun RPC Inspection Overview, page 26-81
- Managing Sun RPC Services, page 26-81
- Verifying and Monitoring Sun RPC Inspection, page 26-82

Sun RPC Inspection Overview

The Sun RPC inspection engine enables or disables application inspection for the Sun RPC protocol. Sun RPC is used by NFS and NIS. Sun RPC services can run on any port. When a client attempts to access an Sun RPC service on a server, it must learn the port that service is running on. It does this by querying the port mapper process, usually rpcbind, on the well-known port of 111.

The client sends the Sun RPC program number of the service and the port mapper process responds with the port number of the service. The client sends its Sun RPC queries to the server, specifying the port identified by the port mapper process. When the server replies, the security appliance intercepts this packet and opens both embryonic TCP and UDP connections on that port.

Note NAT or PAT of Sun RPC payload information is not supported.

Managing Sun RPC Services

Use the Sun RPC services table to control Sun RPC traffic through the security appliance based on established Sun RPC sessions. To create entries in the Sun RPC services table, use the **sunrpc-server** command in global configuration mode:

```
hostname(config)# sunrpc-server interface_name ip_address mask service service_type
protocol {tcp | udp} port[-port] timeout hh:mm:ss
```

You can use this command to specify the timeout after which the pinhole that was opened by Sun RPC application inspection will be closed. For example, to create a timeout of 30 minutes to the Sun RPC server with the IP address 192.168.100.2, enter the following command:

```
hostname(config)# sunrpc-server inside 192.168.100.2 255.255.255.255 service 100003
protocol tcp 111 timeout 00:30:00
```

This command specifies that the pinhole that was opened by Sun RPC application inspection will be closed after 30 minutes. In this example, the Sun RPC server is on the inside interface using TCP port 111. You can also specify UDP, a different port number, or a range of ports. To specify a range of ports, separate the starting and ending port numbers in the range with a hyphen (for example, 111-113).

The service type identifies the mapping between a specific service type and the port number used for the service. To determine the service type, which in this example is 100003, use the **sunrpcinfo** command at the UNIX or Linux command line on the Sun RPC server machine.

To clear the Sun RPC configuration, enter the following command.

```
hostname(config)# clear configure sunrpc-server
```

This removes the configuration performed using the **sunrpc-server** command. The **sunrpc-server** command allows pinholes to be created with a specified timeout.

To clear the active Sun RPC services, enter the following command:

```
hostname(config)# clear sunrpc-server active
```

This clears the pinholes that are opened by Sun RPC application inspection for specific services, such as NFS or NIS.

Verifying and Monitoring Sun RPC Inspection

The sample output in this section is for a Sun RPC server with an IP address of 192.168.100.2 on the inside interface and a Sun RPC client with an IP address of 209.168.200.5 on the outside interface.

To view information about the current Sun RPC connections, enter the **show conn** command. The following is sample output from the **show conn** command:

```
hostname# show conn
15 in use, 21 most used
UDP out 209.165.200.5:800 in 192.168.100.2:2049 idle 0:00:04 flags -
UDP out 209.165.200.5:714 in 192.168.100.2:111 idle 0:00:04 flags -
UDP out 209.165.200.5:712 in 192.168.100.2:647 idle 0:00:05 flags -
UDP out 192.168.100.2:0 in 209.165.200.5:714 idle 0:00:05 flags i
hostname(config)#
```

To display the information about the Sun RPC service table configuration, enter the **show running-config sunrpc-server** command. The following is sample output from the **show running-config sunrpc-server** command:

```
hostname(config)# show running-config sunrpc-server
sunrpc-server inside 192.168.100.2 255.255.255.255 service 100003 protocol UDP port 111
timeout 0:30:00
sunrpc-server inside 192.168.100.2 255.255.255.255 service 100005 protocol UDP port 111
timeout 0:30:00
```

This output shows that a timeout interval of 30 minutes is configured on UDP port 111 for the Sun RPC server with the IP address 192.168.100.2 on the inside interface.

To display the pinholes open for Sun RPC services, enter the **show sunrpc-server active** command. The following is sample output from **show sunrpc-server active** command:

```
hostname# show sunrpc-server active
LOCAL FOREIGN SERVICE TIMEOUT
-------------------------------------------------
1 209.165.200.5/0 192.168.100.2/2049 100003 0:30:00
2 209.165.200.5/0 192.168.100.2/2049 100003 0:30:00
3 209.165.200.5/0 192.168.100.2/647 100005 0:30:00
4 209.165.200.5/0 192.168.100.2/650 100005 0:30:00
```

The entry in the LOCAL column shows the IP address of the client or server on the inside interface, while the value in the FOREIGN column shows the IP address of the client or server on the outside interface.

To view information about the Sun RPC services running on a Sun RPC server, enter the **rpcinfo -p** command from the Linux or UNIX server command line. The following is sample output from the **rpcinfo -p** command:

```
sunrpcserver:~ # rpcinfo -p
program vers proto port
100000 2 tcp 111 portmapper
100000 2 udp 111 portmapper
100024 1 udp 632 status
100024 1 tcp 635 status
100003 2 udp 2049 nfs
100003 3 udp 2049 nfs
100003 2 tcp 2049 nfs
100003 3 tcp 2049 nfs
100021 1 udp 32771 nlockmgr
100021 3 udp 32771 nlockmgr
100021 4 udp 32771 nlockmgr
100021 1 tcp 32852 nlockmgr
100021 3 tcp 32852 nlockmgr
100021 4 tcp 32852 nlockmgr
100005 1 udp 647 mountd
100005 1 tcp 650 mountd
100005 2 udp 647 mountd
100005 2 tcp 650 mountd
100005 3 udp 647 mountd
100005 3 tcp 650 mountd
```

In this output, port 647 corresponds to the mountd daemon running over UDP. The mountd process would more commonly be using port 32780. The mountd process running over TCP uses port 650 in this example.

TFTP Inspection

TFTP inspection is enabled by default.

TFTP, described in RFC 1350, is a simple protocol to read and write files between a TFTP server and client.

The security appliance inspects TFTP traffic and dynamically creates connections and translations, if necessary, to permit file transfer between a TFTP client and server. Specifically, the inspection engine inspects TFTP read request (RRQ), write request (WRQ), and error notification (ERROR).

A dynamic secondary channel and a PAT translation, if necessary, are allocated on a reception of a valid read (RRQ) or write (WRQ) request. This secondary channel is subsequently used by TFTP for file transfer or error notification.

Only the TFTP server can initiate traffic over the secondary channel, and at most one incomplete secondary channel can exist between the TFTP client and server. An error notification from the server closes the secondary channel.

TFTP inspection must be enabled if static PAT is used to redirect TFTP traffic.

XDMCP Inspection

XDMCP inspection is enabled by default; however, the XDMCP inspection engine is dependent upon proper configuration of the **established** command.

XDMCP is a protocol that uses UDP port 177 to negotiate X sessions, which use TCP when established.

For successful negotiation and start of an XWindows session, the security appliance must allow the TCP back connection from the Xhosted computer. To permit the back connection, use the **established** command on the security appliance. Once XDMCP negotiates the port to send the display, The **established** command is consulted to verify if this back connection should be permitted.

During the XWindows session, the manager talks to the display Xserver on the well-known port 6000 | n. Each display has a separate connection to the Xserver, as a result of the following terminal setting.

```
setenv DISPLAY Xserver:n
```

where *n* is the display number.

When XDMCP is used, the display is negotiated using IP addresses, which the security appliance can NAT if needed. XDCMP inspection does not support PAT.



CHAPTER 27

Configuring Cisco Unified Communications Proxy Features

This chapter describes how to configure the adaptive security appliance for Cisco Unified Communications Proxy features.

This chapter includes the following sections:

- Overview of the Adaptive Security Appliance in Cisco Unified Communications, page 27-1
- TLS Proxy Applications in Cisco Unified Communications, page 27-3
- Phone Proxy, page 27-5
- TLS Proxy for Encrypted Voice Inspection, page 27-42
- Cisco Unified Mobility and MMP Inspection Engine, page 27-52
- Cisco Unified Presence, page 27-59
- Sample Configurations for Cisco Unified Communications Proxy Features, page 27-65

Overview of the Adaptive Security Appliance in Cisco Unified Communications

This section describes the Cisco UC Proxy features on the Cisco ASA 5500 series appliances. The purpose of a proxy is to terminate and reoriginate connections between a client and server. The proxy delivers a range of security functions such as traffic inspection, protocol conformance, and policy control to ensure security for the internal network. An increasingly popular function of a proxy is to terminate encrypted connections in order to apply security policies while maintaining confidentiality of connections. The Cisco ASA 5500 Series appliances are a strategic platform to provide proxy functions for unified communications deployments.

The Cisco UC Proxy includes the following solutions:

Phone Proxy: Secure remote access for Cisco encrypted endpoints, and VLAN traversal for Cisco softphones

The phone proxy feature enables termination of Cisco SRTP/TLS-encrypted endpoints for secure remote access. The phone proxy allows large scale deployments of secure phones without a large scale VPN remote access hardware deployment. End-user infrastructure is limited to just the IP endpoint, without VPN tunnels or hardware.

The Cisco adaptive security appliance phone proxy is the replacement product for the Cisco Unified Phone Proxy. Additionally, the phone proxy can be deployed for voice/data VLAN traversal for softphone applications. Cisco IP Communicator (CIPC) traffic (both media and signaling) can be proxied through the security appliance, thus traversing calls securely between voice and data VLANs.

For information about the differences between the TLS proxy and phone proxy, go to the following URL for Unified Communications content, including TLS Proxy vs. Phone Proxy white paper:

http://www.cisco.com/go/secureuc

TLS Proxy: Decryption and inspection of Cisco Unified Communications encrypted signaling

End-to-end encryption often leaves network security appliances "blind" to media and signaling traffic, which can compromise access control and threat prevention security functions. This lack of visibility can result in a lack of interoperability between the firewall functions and the encrypted voice, leaving businesses unable to satisfy both of their key security requirements.

The security appliance is able to intercept and decrypt encrypted signaling from Cisco encrypted endpoints to the Cisco Unified Communications Manager (Cisco UCM), and apply the required threat protection and access control. It can also ensure confidentiality by re-encrypting the traffic onto the Cisco UCM servers.

Typically, the security appliance TLS Proxy functionality is deployed in campus unified communications network. This solution is ideal for deployments that utilize end to end encryption and firewalls to protect Unified Communications Manager servers.

Mobility Proxy: Secure connectivity between Cisco Unified Mobility Advantage server and Cisco Unified Mobile Communicator clients

Cisco Unified Mobility solutions include the Cisco Unified Mobile Communicator (Cisco UMC), an easy-to-use software application for mobile handsets that extends enterprise communications applications and services to mobile phones and the Cisco Unified Mobility Advantage (Cisco UMA) server. The Cisco Unified Mobility solution streamlines the communication experience, enabling single number reach and integration of mobile endpoints into the Unified Communications infrastructure.

The security appliance acts as a proxy, terminating and reoriginating the TLS signaling between the Cisco UMC and Cisco UMA. As part of the proxy security functionality, inspection is enabled for the Cisco UMA Mobile Multiplexing Protocol (MMP), the protocol between Cisco UMC and Cisco UMA.

Presence Federation Proxy: Secure connectivity between Cisco Unified Presence servers and Cisco/Microsoft Presence servers

Cisco Unified Presence solution collects information about the availability and status of users, such as whether they are using communication devices, such as IP phones at particular times. It also collects information regarding their communications capabilities, such as whether web collaboration or video conferencing is enabled. Using user information captured by Cisco Unified Presence, applications such as Cisco Unified Personal Communicator and Cisco UCM can improve productivity by helping users connect with colleagues more efficiently through determining the most effective way for collaborative communication.

Using the security appliance as a secure presence federation proxy, businesses can securely connect their Cisco Unified Presence (Cisco UP) servers to other Cisco or Microsoft Presence servers, enabling intra-enterprise communications. The security appliance terminates the TLS connectivity between the servers, and can inspect and apply policies for the SIP communications between the servers.

TLS Proxy Applications in Cisco Unified Communications

Table 27-1 shows the Cisco Unified Communications applications that utilize the TLS proxy on the security appliance.

Table 27-1 TLS Proxy Applications and the Security Appliance

Application	TLS Client	TLS Server	Client Authentication	Security Appliance Server Role	Security Appliance Client Role
Phone Proxy and TLS Proxy	IP phone	Cisco UCM	Yes	Proxy certificate, self-signed or by internal CA	Local dynamic certificate signed by the security appliance CA (might not need certificate for phone proxy application)
Mobility Proxy	Cisco UMC	Cisco UMA	No	Using the Cisco UMA private key or certificate impersonation	Any static configured certificate
Presence Federation Proxy	Cisco UP or MS LCS/OCS	Cisco UP or MS LCS/OCS	Yes	Proxy certificate, self-signed or by internal CA	Using the Cisco UP private key or certificate impersonation

The security appliance supports TLS proxy for various voice applications. For the phone proxy, the TLS proxy running on the security appliance has the following key features:

- The security appliance forces remote IP phones connecting to the phone proxy through the Internet to be in secured mode even when the Cisco UCM cluster is in non-secure mode.
- The TLS proxy is implemented on the security appliance to intercept the TLS signaling from IP phones.
- The TLS proxy decrypts the packets, sends packets to the inspection engine for NAT rewrite and protocol conformance, optionally encrypts packets, and sends them to Cisco UCM or sends them in clear text if the IP phone is configured to be in nonsecure mode on the Cisco UCM.
- The security appliance acts as a media terminator as needed and translates between SRTP and RTP media streams.
- The TLS proxy is a transparent proxy that works based on establishing trusted relationship between the TLS client, the proxy (the security appliance), and the TLS server.

For the Cisco Unified Mobility solution, the TLS client is a Cisco UMA client and the TLS server is a Cisco UMA server. The security appliance is between a Cisco UMA client and a Cisco UMA server. The mobility proxy (implemented as a TLS proxy) for Cisco Unified Mobility allows the use of an imported PKCS-12 certificate for server proxy during the handshake with the client. Cisco UMA clients are not required to present a certificate (no client authentication) during the handshake.

For the Cisco Unified Presence solution, the security appliance acts as a TLS proxy between the Cisco UP server and the foreign server. This allows the security appliance to proxy TLS messages on behalf of the server that initiates the TLS connection, and route the proxied TLS messages to the client. The security appliance stores certificate trustpoints for the server and the client, and presents these certificates on establishment of the TLS session.

Licensing for Cisco Unified Communications Proxy Features

The Cisco Unified Communications proxy features supported by the security appliance require a Unified Communications Proxy license:

- Phone proxy
- TLS proxy for encrypted voice inspection
- Mobility proxy
- Presence federation proxy

The Unified Communications proxy features are licensed by TLS session. For the phone proxy or TLS proxy, each IP phone may have a single connection to the Cisco UCM server or two connections —one connection to the primary Cisco UCM and one connection to the backup Cisco UCM. In the second scenario, the phone proxy uses two Unified Communications Proxy sessions because two TLS sessions are set up. For the mobility proxy and presence federation proxy, each endpoint utilizes one Unified Communications Proxy session.

Table 27-2 shows the Unified Communications Proxy license details by platform.

Table 27-2 License Requirements for the Security Appliance

Security Appliance Platform	Max UC Proxy Licenses	Tiers for UC Proxy Licenses
ASA 5505	24	24
ASA 5510	100	24, 50, 100
ASA 5520	1,000	24, 50, 100, 250, 500, 750, 1000
ASA 5540	2,000	24, 50, 100, 250, 500, 750, 1000, 2000
ASA 5550	3,000	24, 50, 100, 250, 500, 750, 1000, 2000, 3000

A Unified Communications Proxy license is applied the same way as other licensed features (such as, SSL VPN), via the **activation-key** command. To check the license on the security appliance, use the **show version** or **show activation-key** command:

```
hostname# show activation-key
Serial Number:  P3000000179
Running Activation Key: 0xa700d24c 0x98caab35 0x88038550 0xaf383078 0x02382080

Licensed features for this platform:
Maximum Physical Interfaces  : Unlimited
Maximum VLANs                : 150
Inside Hosts                 : Unlimited
Failover                     : Active/Active
VPN-DES                      : Enabled
VPN-3DES-AES                 : Enabled
Security Contexts            : 10
GTP/GPRS                     : Enabled
VPN Peers                    : 750
```

```
WebVPN Peers                  : 750
AnyConnect for Mobile         : Disabled
AnyConnect for Linksys phone  : Disabled
Advanced Endpoint Assessment  : Enabled
UC Proxy Sessions             : 1000
This platform has an ASA 5520 VPN Plus license.

The flash activation key is the SAME as the running key.
hostname#
```

See the following links for additional information on licensing. If you are a registered user of Cisco.com and would like to obtain a Unified Communications Proxy license, go to the following website:

http://www.cisco.com/go/license

If you are not a registered user of Cisco.com, go to the following website:

https://tools.cisco.com/SWIFT/Licensing/RegistrationServlet

Provide your name, e-mail address, and the serial number for the security appliance as it appears in the show version command output.

Phone Proxy

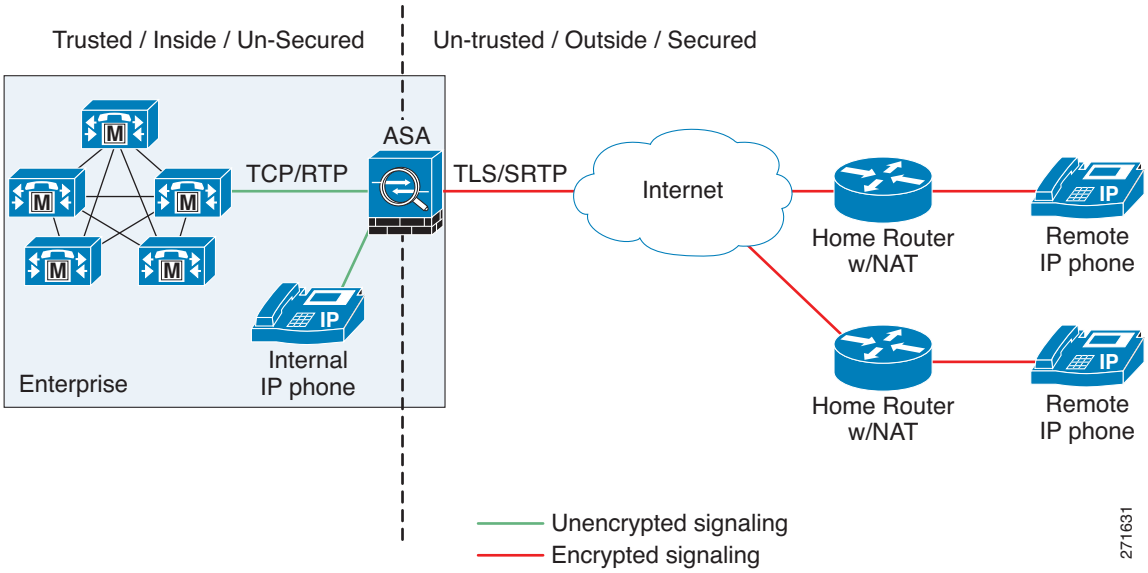
This section includes the following topics:

- About the Phone Proxy, page 27-5
- Phone Proxy Configuration, page 27-8
- Troubleshooting the Phone Proxy, page 27-26

About the Phone Proxy

The phone proxy on the security appliance bridges IP telephony between the corporate IP telephony network and the Internet in a secure manner by forcing data from remote phones on an untrusted network to be encrypted. Telecommuters can connect their IP phones to the corporate IP telephony network over the Internet securely via the phone proxy without the need to connect over a VPN tunnel as illustrated by Figure 27-1.

Figure 27-1 Phone Proxy Secure Deployment



The phone proxy supports a Cisco UCM cluster in mixed mode or nonsecure mode. Regardless of the cluster mode, the remote phones that are capable of encryption are always forced to be in encrypted mode. TLS (signaling) and SRTP (media) are always terminated on the security appliance. The security appliance can also perform NAT, open pinholes for the media, and apply inspection policies for the SCCP and SIP protocols. In a nonsecure cluster mode or a mixed mode where the phones are configured as nonsecure, the phone proxy behaves in the following ways:

- The TLS connections from the phones are terminated on the security appliance and a TCP connection is initiated to the Cisco UCM.
- SRTP sent from external IP phones to the internal network IP phone via the security appliance is converted to RTP.

In a mixed mode cluster where the internal IP phones are configured as authenticated, the TLS connection is not converted to TCP to the Cisco UCM but the SRTP is converted to RTP.

In a mixed mode cluster where the internal IP phone is configured as encrypted, the TLS connection remains a TLS connection to the Cisco UCM and the SRTP from the remote phone remains SRTP to the internal IP phone.

Since the main purpose of the phone proxy is to make the phone behave securely while making calls to a nonsecure cluster, the phone proxy performs the following major functions:

- Creates the certificate trust list (CTL) file, which is used to perform certificate based authentication with remote phones.
- Modifies the IP phone configuration file when it is requested via TFTP, changes security fields from nonsecure to secure, and signs all files sent to the phone. These modifications secure remote phones by forcing the phones to perform encrypted signaling and media.
- Terminates TLS signaling from the phone and initiates TCP or TLS to Cisco UCM
- Inserts itself into the media path by modifying the Skinny and SIP signaling messages.
- Terminates SRTP and initiates RTP/SRTP to the called party.

Note As an alternative to authenticating remote IP phones through the TLS handshake, you can configure authentication via LSC provisioning. With LSC provisioning you create a password for each remote IP phone user and each user enters the password on the remote IP phones to retrieve the LSC.

Because using LSC provisioning to authenticate remote IP phones requires the IP phones first register in nonsecure mode, Cisco recommends LSC provisioning be done inside the corporate network before giving the IP phones to end-users. Otherwise, having the IP phones register in nonsecure mode requires the Administrator to open the nonsecure signaling port for SIP and SCCP on the security appliance.

See "Example 5: LSC Provisioning in Mixed-mode Cisco UCM cluster; Cisco UCM and TFTP Server on Publisher, page 27-71". See also the Cisco Unified Communications Manager Security Guide for information on Using the Certificate Authority Proxy Function (CAPF) to install a locally significant certificate (LSC).

Phone Proxy Limitations and Restrictions

The phone proxy has the following limitations and restrictions:

- The phone proxy is not supported in multiple context mode.
- The phone proxy does not support inspection of packets from phones connecting to the phone proxy over a VPN tunnel. Therefore, sending phone proxy traffic through a VPN tunnel is not supported. Configuring the phone proxy feature on the security appliance allows IP phones to connect to the corporate network without requiring that the traffic go through VPN tunnels.
- The phone proxy does not support recording calls when the recording traffic must traverse the security appliance to get to the recording device. For example, the Unified Communication Manager versions 6.x and 7.x supports using a third-party recording device with the forking feature. When the recording feature is used with the phone proxy, the feature creates a second RTP media stream that is a copy of the original RTP media stream. The existence of two RTP media streams from the outside IP phone to the recording device on behind the security device disrupts the IP phone audio.
- The security appliance supports stateful failover for the phone proxy in the following way. When the active unit goes down, any calls from IP phones going through the phone proxy fail, media stops flowing, and the IP phones should unregister from the failed unit and reregister with the active unit. Then, the calls must be re-established."
- The phone proxy is not supported when the security appliance is running in transparent mode or multimode.
- The phone proxy does not support communication with internal IP phones that natively use the Secure Real-Time Protocol (SRTP).
- The phone proxy does not support IP phones sending Real-Time Control Protocol (RTCP) packets through the security appliance. Disable RTCP packets in the Cisco Unified CM Administration console from the Phone Configuration page. See your Cisco Unified Communications Manager (CallManager) documentation for information about setting this configuration option.
- When used with CIPC, the phone proxy does not support end-users resetting their device name in CIPC (Prefences > Network tab > Use this Device Name field) or Administrators resetting the devide name in Cisco Unified CM Administration console (Device menu > Phone Configuration > Device Name field). To function with the phone proxy, the CIPC configuration file must be in the format: SEP<mac_address>.cnf.xml. If the device name does not follow this format (SEP<mac_address>), CIPC cannot retrieve its configuration file from Cisco UMC via the phone proxy and CIPC will not function.

- The phone proxy does not support IP phones sending SCCP video messages using Cisco VT Advantage because SCCP video messages do not support SRTP keys.
- For mixed-mode clusters, the phone proxy does not support the Cisco Unified Call Manager using TFTP to send encrypted configuration files to IP phones through the security appliance.
- The phone proxy and CIPC are not supported when CIPC is installed on computers in remote locations, such that the calls from those computers traverse the Internet, terminating at the security appliance, to reach IP phones residing on the network behind the adaptive security appliance. The computers where CIPC is installed must be on the network to reach the IP phones behind the security appliance.
- Multiple IP phones behind one NAT device must be configured to use the same security mode.

  When the phone proxy is configured for a mixed-mode cluster and multiple IP phones are behind one NAT device and registering through the phone proxy, all the SIP and SCCP IP phones must be configured as authenticated or encrypted, or all as non-secure on the Unified Call Manager.

  For example, if there are four IP phones behind one NAT device where two IP phones are configured using SIP and two IP phones are configured using SCCP, the following configurations on the Unified Call Manager are acceptable:

  – Two SIP IP phones: one IP phone in authenticated mode and one in encrypted mode, both in authenticated mode, or both in encrypted mode

    Two SCCP IP phones: one IP phone in authenticated mode and one in encrypted mode, both in authenticated mode, or both in encrypted mode

  – Two SIP IP phones: both in non-secure mode

    Two SCCP IP phones: one IP phone in authenticated mode and one in encrypted mode, both in authenticated mode, both in encrypted mode

  – Two SIP IP phones: one IP phone in authenticated mode and one in encrypted mode, both in authenticated mode, both in encrypted mode

    Two SCCP IP phones: both in non-secure mode

  This limitation results from the way the application-redirect rules (rules that convert TLS to TCP) are created for the IP phones.

Phone Proxy Configuration

This section includes the following topics:

- Configuration Prerequisites, page 27-9
- Requirements to Support the 7960 and 7940 IP Phones, page 27-11
- Addressing Requirements for IP Phones on Multiple Interfaces, page 27-11
- Supported Cisco UCM and IP Phones for the Phone Proxy, page 27-12
- End-User Phone Provisioning, page 27-13
- Configuring the Phone Proxy in a Non-secure Cisco UCM Cluster, page 27-13
- Importing Certificates from the Cisco UCM, page 27-17
- Configuring the Phone Proxy in a Mixed-mode Cisco UCM Cluster, page 27-19
- Phone Proxy Configuration for Cisco IP Communicator, page 27-24
- Configuring Linksys Routers for UDP Port Forwarding, page 27-24

- About Rate Limiting TFTP Requests, page 27-25
- About ICMP Traffic Destined for the Media Termination Address, page 27-26

Configuration Prerequisites

Before configuring the phone proxy, ensure that the security appliance meets the following configuration requirements:

- The security appliance must have an IP address for media termination that meets the following criteria:
  - The IP address is a publicly routable address that is an unused IP address within an address range associated with the outside network interface on the security appliance.
  - The IP address cannot be the same address of an interface on the security appliance. This includes using the IP address of the external interface on the security appliance to which remote IP phones connect.
  - The IP address cannot overlap with existing static NAT pools or NAT rules.
  - The IP address cannot be the same as the Cisco UCM or TFTP server IP address.
  - For IP phones behind a router or gateway, add routes to the media termination address on the router or gateway so that the phone can reach the media termination address.

Note If your organization security policy dictates that IP phones on internal networks must not have routes to external networks, we recommended that you use a Unified-Communications-aware NAT device on the internal network. By representing the media termination address with an address within the internal network address range, you do not need to expose the internal IP phones to external routes.

- The TFTP server must reside on the same interface as the Cisco UCM.
- If you have an fully qualified domain name (FQDN) configured for the Cisco UCM rather than an IP address, you must configure and enable DNS lookup on the security appliance. For information about the **dns domain-lookup** command and how to use it to configure DNS lookup, see *Cisco Security Appliance Command Reference*.

  After configuring the DNS lookup, make sure that the security appliance can ping the Cisco UCM with the configured FQDN.

  If you have a CAPF service enabled and the Cisco UCM is not running on the Publisher, and the Publisher is configured with a FQDN instead of an IP address, you must also configure DNS lookup.
- Access-list rules must be configured to allow TFTP requests.

  Table 27-3 lists the access-list rule that must be configured for TFTP on the security appliance:

Table 27-3 Access List Rule for TFTP

Address	Port	Protocol	Description
TFTP Server	69	UDP	Allow incoming TFTP

Note 3804 is the default value for the CAPF Service. This default value should be modified if it is modified on the Cisco UCM. If NAT is configured for the TFTP server or Cisco UCMs, the translated "global" address must be used in the access lists.

For information about configuring access-lists on the security appliance, see Access List Overview, page 18-1.

- If the phone proxy is deployed behind an existing firewall, access-list rules to permit signaling, TFTP, and media traffic to the phone proxy must be configured. If NAT is required for Cisco UCM, it must be configured on the security appliance, not on the existing firewall.

  Table 27-4 lists the ports that are required to be configured on the existing firewall:

Table 27-4 Port Configuration Requirements

Address	Port	Protocol	Description
Media Termination	1024-65535	UDP	Allow incoming SRTP
TFTP Server	69	UDP	Allow incoming TFTP
Cisco UCM	2443	TCP	Allow incoming secure SCCP
Cisco UCM	5061	TCP	Allow incoming secure SIP
CAPF Service (on Cisco UCM)	3804	TCP	Allow CAPF service for LSC provisioning

Note All these ports are configurable on the Cisco UCM, except for TFTP. These are the default values and should be modified if they are modified on the Cisco UCM. If NAT is configured for the TFTP server or Cisco UCMs, the translated "global" address must be used in the access lists.

- If NAT is configured for the TFTP server, the NAT configuration must be configured prior to configuring the **tftp-server** command under the phone proxy.
- The Cisco UCM can be on a private network on the inside but you need to have a static mapping for the Cisco UCM on the security appliance to a public routable address.
- The following PAT configuration requirements must be met for the phone proxy:
  - When the Skinny inspection global port is configured to use a non-default port, then you must configure the nonsecure port as the `global_sccp_port+443`.

    Therefore, if *global_sccp_port* is 7000, then the global secure SCCP port is 7443. Reconfiguring the port might be necessary when the phone proxy deployment has more than one Cisco UCM and they must share the interface IP address or a global IP address:

```
/* use the default ports for the first CUCM */
static (inside,outside) tcp interface 2000 10.0.0.1 2000
static (inside,outside) tcp interface 2443 10.0.0.1 2443
/* use non-default ports for the 2nd CUCM */
static (inside,outside) tcp interface 7000 10.0.0.2 2000
static (inside,outside) tcp interface 7443 10.0.0.2 2443
```

Note Both PAT configurations—for the nonsecure and secure ports—must be configured.

– When the IP phones must contact the CAPF on the Cisco UCM and the Cisco UCM is configured with static PAT (LCS provisioning is required), you must configure static PAT for the default CAPF port 3804.

Requirements to Support the 7960 and 7940 IP Phones

To support the 7960 and 7940 IP phones with the phone proxy, you must meet the following requirements:

- An LSC must be installed on these IP phones because they do not come pre installed with a MIC. Install the LSC on each phone before using them with the phone proxy to avoid opening the nonsecure SCCP port for the IP phones to register in nonsecure mode with the Cisco UCM.

  See the following document for the steps to install an LSC on IP phones:

  http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/security/7_0_1/secugd/secucapf.html#wp1093518

Note If an IP phone already has an LSC installed on it from a different Cisco UCM cluster, delete the LSC from the different cluster and install an LSC from the current Cisco UCM cluster.

- The CAPF certificate must be imported onto the security appliance.
- The CTL file created on the security appliance must be created with a CAPF record-entry.
- The phone must be configured to use only the SCCP protocol because the SIP protocol does not support encryption on these IP phones.
- If LSC provisioning is done via the phone proxy, you must add an ACL to allow the IP phones to register with the Cisco UCM on the nonsecure port 2000.

Addressing Requirements for IP Phones on Multiple Interfaces

When IP phones reside on multiple interfaces, the phone proxy configuration must have the correct IP address set for the Cisco UCM in the CTL file.

See the following example topology for information about how to correctly set the IP address:

```
phones --- (dmz)-----|
                     |----- ASA PP --- (outside Internet) --- phones
phones --- (inside)--|
```

In this example topology, the following IP address are set:

- Cisco UCM on the inside interface is set to 10.0.0.5
- The DMZ network is 192.168.1.0/24
- The inside network is 10.0.0.0/24

The Cisco UCM is mapped with different global IP addresses from DMZ > outside and inside interfaces > outside interface.

In the CTL file, the Cisco UCM must have two entries because of the two different IP addresses. For example, if the static statements for the Cisco UCM are as follows:

```
static (inside,outside) 128.106.254.2 10.0.0.5
static (inside,dmz) 192.168.1.2 10.0.0.5
```

There must be two CTL file record entries for the Cisco UCM:

```
record-entry cucm trustpoint cucm_in_to_out address 128.106.254.2
record-entry cucm trustpoint cucm_in_to_dmz address 192.168.1.2
```

Supported Cisco UCM and IP Phones for the Phone Proxy

Cisco Unified Communications Manager

The following release of the Cisco Unified Communications Manager are supported with the phone proxy:

- Cisco Unified CallManager Version 4.x
- Cisco Unified CallManager Version 5.x
- Cisco Unified Communications Manager 6.x
- Cisco Unified Communications Manager 7.x

Cisco Unified IP Phones

The following IP phones in the Cisco Unified IP Phones 7900 Series are supported with the phone proxy:

- Cisco Unified IP Phone 7975
- Cisco Unified IP Phone 7971
- Cisco Unified IP Phone 7970
- Cisco Unified IP Phone 7965
- Cisco Unified IP Phone 7962
- Cisco Unified IP Phone 7961
- Cisco Unified IP Phone 7961G-GE
- Cisco Unified IP Phone 7960 (SCCP protocol support only)
- Cisco Unified IP Phone 7945
- Cisco Unified IP Phone 7942
- Cisco Unified IP Phone 7941
- Cisco Unified IP Phone 7941G-GE
- Cisco Unified IP Phone 7940 (SCCP protocol support only)
- Cisco Unified Wireless IP Phone 7921
- CIPC for softphones (CIPC versions with Authenticated mode only)

Note The Cisco IP Communicator is supported with the phone proxy VLAN Traversal in authenticated TLS mode.

The phone proxy and CIPC are not supported when CIPC is installed on computers in remote locations, such that the calls from those computers traverse the Internet, terminating at the security appliance, to reach IP phones residing on the network behind the adaptive security appliance. The computers where CIPC is installed must be on the network to reach the IP phones behind the security appliance.

End-User Phone Provisioning

The phone proxy is a transparent proxy with respect to the TFTP and signaling transactions. If NAT is not configured for the Cisco UCM TFTP server, then the phone needs to be configured with the Cisco UCM cluster TFTP server address.

If NAT is configured for the Cisco UCM TFTP server, then the Cisco UCM TFTP server global address is configured as the TFTP server address on the phone.

- Option 1 (Recommended) – Stage the IP phones at corporate headquarters before sending them to the end users:
  - The phones register inside the network. IT ensures there are no issues with the phone configurations, image downloads, and registration.
  - If Cisco UCM cluster was in mixed mode, the CTL file should be erased before sending the phone to the end user.
  - Advantages of this option are:

    • Easier to troubleshoot and isolate problems with the network or phone proxy because you know whether the phone is registered and working with the Cisco UCM.

    • Better user experience because the phone does not have to download firmware from over a broadband connection, which can be slow and require the user to wait for a longer time.
- Option 2 – Send the new phone to the end user
  - The user must be provided instructions to change the settings on phones with the appropriate Cisco UCM and TFTP server IP address.

In both options, deploying a remote IP phone behind a commercial Cable/DSL router with NAT capabilities is supported.

Note As an alternative to authenticating remote IP phones through the TLS handshake, you can configure authentication via LSC provisioning. With LSC provisioning you create a password for each remote IP phone user and each user enters the password on the remote IP phones to retrieve the LSC.

Because using LSC provisioning to authenticate remote IP phones requires the IP phones first register in nonsecure mode, Cisco recommends LSC provisioning be done inside the corporate network before giving the IP phones to end-users. Otherwise, having the IP phones register in nonsecure mode requires the Administrator to open the nonsecure signaling port for SIP and SCCP on the security appliance.

See "Example 5: LSC Provisioning in Mixed-mode Cisco UCM cluster; Cisco UCM and TFTP Server on Publisher, page 27-71". See also the Cisco Unified Communications Manager Security Guide for information on Using the Certificate Authority Proxy Function (CAPF) to install a locally significant certificate (LSC).

Configuring the Phone Proxy in a Non-secure Cisco UCM Cluster

Step 1 Create trustpoints and generate certificates for each entity in the network (Cisco UCM, Cisco UCM and TFTP, TFTP server, CAPF) that the IP phone must trust. The certificates are used in creating the CTL file.

You need to create trustpoints for each Cisco UCM (primary and secondary if a secondary Cisco UCM is used) and TFTP server in the network. The trustpoints need to be in the CTL file for the phones to trust the Cisco UCM.

a. Create a keypair that can be used for the trustpoints by entering the following command:

```
hostname(config)# crypto key generate rsa label key-pair-label modulus size
```

b. Create the trustpoints for each Cisco UCM (primary and secondary) by entering the following commands:

```
hostname(config)# crypto ca trustpoint trustpoint_name
hostname(config-ca-trustpoint)# enrollment self
hostname(config-ca-trustpoint)# keypair keyname
hostname(config-ca-trustpoint)# exit
hostname(config)# crypto ca enroll trustpoint
```

Entering these commands generates a self-signed certificate and specifies the keypair whose public key is being certified. This is the keypair created in substep a. Entering the **crypto ca enroll** command requests the certificate from the CA server and causes the security appliance to generate the certificate.

c. Create the trustpoint for the TFTP server by entering the following commands:

```
hostname(config)# crypto ca trustpoint trustpoint_name
hostname(config-ca-trustpoint)# enrollment self
hostname(config-ca-trustpoint)# keypair keyname
hostname(config-ca-trustpoint)# exit
hostname(config)# crypto ca enroll trustpoint
```

Note You are only required to perform this step when the TFTP server resides on a different server from the Cisco UCM. See Example 3: Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Different Servers, page 27-68 for an example of this configuration.

d. When prompted to include the device serial number in the subject name, type **Y** to include the serial number or type **N** to exclude it.

e. When prompted to generate the self-signed certificate, type **Y**.

f. Import the following certificates which are stored on the Cisco UCM. These certificates are required by the security appliance for the phone proxy.

- Cisco_Manufacturing_CA
- CAP-RTP-001
- CAP-RTP-002

See Importing Certificates from the Cisco UCM, page 27-17. For example, the CA Manufacturer certificate is required by the phone proxy to validate the IP phone certificate.

g. (Optional) If LSC provisioning is required or you have LSC enabled IP phones, you must import the CAPF certificate from the Cisco UCM. See Importing Certificates from the Cisco UCM, page 27-17.

Note If the Cisco UCM has more than one CAPF certificate, you must import all of them to the security appliance.

Step 2 Create the CTL file that will be presented to the IP phones during the TFTP. The *address* here must be the translated or global address of the TFTP server or Cisco UCM if NAT is configured.

a. If you are using domain names for your Cisco UCM and TFTP server, you must configure DNS lookup on the security appliance. Add an entry for each of the outside interfaces on the security appliance into your DNS server, if such entries are not already present. Each security appliance outside IP address should have a DNS entry associated with it for lookups. These DNS entries must also be enabled for Reverse Lookup. Enable DNS lookups on your security appliance with the **dns domain-lookup** *interface_name* command (where the *interface_name* specifies the interface that has a route to your DNS server). Additionally, define your DNS server IP address on the security appliance; for example: `dns name-server 10.2.3.4` (IP address of your DNS server).



Note You can enter the **dns domain-lookup** command multiple times to enable DNS lookup on multiple interfaces. If you enter multiple commands, the security appliance tries each interface in the order it appears in the configuration until it receives a response.

b. Create the CTL file instance by entering the following command:

```
hostname(config)# ctl-file ctl_name
```

c. Create the record entry for the TFTP server by entering the following command. Use the global or mapped IP address of the TFTP server.

```
hostname(config-ctl-file)# record-entry tftp trustpoint trustpoint_name address
TFTP_IP_address
```

d. Create the record entry for the each Cisco UCM (primary and secondary) by entering the following command. Use the global or mapped IP address of the Cisco UCM.

```
hostname(config-ctl-file)# record-entry cucm trustpoint trustpoint_name address
IP_address
```

e. (Optional) If LSC provisioning is or you have LSC enabled IP phones, create the record entry for CAPF by entering the following command:

```
hostname(config-ctl-file)# record-entry capf trustpoint trust_point address
```

f. Create the CTL file by entering the following command:

```
hostname(config-ctl-file)# no shutdown
```

When the file is created, it creates an internal trustpoint used by the phone proxy to sign the TFTP files. The trustpoint is named **_internal_PP**_*ctl-instance_filename*.

g. Save the certificate configuration to Flash memory by entering the following command:

```
hostname(config)# copy running-configuration startup-configuration
```

Step 3 Create the TLS proxy instance to handle the encrypted signaling.

a. Create the TLS proxy instance by entering the following command:

```
hostname(config)# tls-proxy proxy_name
```

b. Configure the server trustpoint and reference the internal trustpoint named **_internal_PP**_*ctl-instance_filename*:

```
hostname(config-tlsp)# server trust-point _internal_PP_ctl-instance_filename
```

Step 4 Configure the phone proxy instance.

a. Create the CTL file instance:

```
hostname(config)# phone-proxy phone_proxy_name
```

b. Configure the media-termination address used by the phone-proxy for SRTP and RTP by entering the following command:

```
hostname(config-phone-proxy)# media-termination address ip_address
```

Note

- For the media termination address, you must select a publicly routable IP address that is an unused IP address on an attached outside network to the security appliance interface that will never be used by another device in your network.
- Specifically, the media termination address cannot be the same as any security appliance interface IP address, cannot overlap with existing static NAT rules, and cannot be the same as the Cisco UCM or TFTP server IP address.
- For IP phones behind a router or gateway, add routes to the media termination address on the router or gateway so that the phone can reach the media termination address.

c. Create the TFTP server using the actual internal address and specify the interface on which the TFTP server resides by entering the following command:

```
hostname(config-phone-proxy)# tftp-server address ip_address interface interface
```

d. Configure the TLS proxy instance created in Step 3 by entering the following command:

```
hostame(config-phone-proxy)# tls-proxy proxy_name
```

e. Configure the CTL file instance created in Step 2 by entering the following command:

```
hostname(config-phone-proxy)# ctl-file ctl_name
```

f. (Optional) If the operational environment has an external HTTP proxy to which the IP phones direct all HTTP request, enter the following command to configure a proxy server on the security appliance:

```
hostname(config-phone-proxy)# proxy-server address ip_address [listen_port] interface
ifc
```

You can configure only one proxy server while the phone proxy is in use; however, if the IP phones have already downloaded their configuration files after you have configured the proxy server, you must restart the IP phones so that they get the configuration file with the proxy server address in the file.

By default, the Phone URL Parameters configured under the Enterprise Parameters use an FQDN in the URLs. The parameters might need to be changed to use an IP address if the DNS lookup for the HTTP proxy does not resolve the FQDNs.

g. (Optional) To force Cisco IP Communicator (CIPC) softphones to operate in authenticated mode when CIPC softphones are deployed in a voice and data VLAN scenario, enter the following command:

```
hostname(config-phone-proxy)# cipc security-mode authenticated
```

See Phone Proxy Configuration for Cisco IP Communicator, page 27-24 for all requirements for using the phone proxy with CIPC.

h. (Optional) To preserve the settings configured on the Cisco UCM for each IP phone configured, enter the following command:

```
hostname(config-phone-proxy)# no disable service-settings
```

By default, the following settings are disabled on the IP phones:

- PC Port
- Gratuitous ARP
- Voice VLAN access
- Web Access
- Span to PC Port

Step 5 Enable the phone proxy with SIP and Skinny inspection.

a. Configure the secure Skinny class of traffic to inspect by entering the following commands:

```
hostname(config)# class-map class_map_name
hostname(config-cmap)# match port tcp eq 2443
```

Where *class_map_name* is the name of the Skinny class map.

b. Configure the secure SIP class of traffic to inspect by entering the following commands:

```
hostname(config)# class-map class_map_name
hostname(config-cmap)# match port tcp eq 5061
```

Where *class_map_name* is the name of the SIP class map.

c. Configure the policy map and attach the action to the class of traffic by entering the following commands:

```
hostname(config)# policy-map name
hostname(config-pmap)# class classmap-name
hostname(config-pmap-c)# inspect skinny phone-proxy pp_name
hostnae(config-pmap)# class classmap-name
hostname(config-pmap-c)# inspect sip phone-proxy pp_name
```

Where *classmap_name* is the name of the Skinny class map and the name for the SIP class map.

d. Enable the policy on the outside interface by entering the following command:

```
hostname(config)# service-policy policymap_name interface intf
```

Importing Certificates from the Cisco UCM

For the TLS proxy used by the phone proxy to complete the TLS handshake successfully, it needs to verify the certificates from the IP phone (and the Cisco UCM if doing TLS with Cisco UCM). To validate the IP phone certificate, we need the CA Manufacturer certificate which is stored on the Cisco UCM. Follow these steps to import the CA Manufacturer certificate to the security appliance.

Step 1 Go to the Cisco UCM Operating System Administration web page.

Step 2 Choose **Security > Certificate Management**.

Note Earlier versions of Cisco UCM have a different UI and way to locate the certificates. For example, in Cisco UCM version 4.x, certificates are located in the directory `C:\Program Files\Cisco\Certificates`. See your Cisco Unified Communications Manager (CallManager) documentation for information about locating certificates.

Step 3 Click Find and it will display all the certificates.

Step 4 Find the filename `Cisco_Manufacturing_CA`. This is the certificate need to verify the IP phone certificate. Click the .PEM file `Cisco_Manufacturing_CA.pem`. This will show you the certificate information and a dialog box that has the option to download the certificate.

Note If the certificate list contains more than one certificate with the filename `Cisco_Manufacturing_CA`, make you select the certificate `Cisco_Manufacturing_CA.pem`—the one with the .pem file extension.

Step 5 Click Download and save the file as a text file.

Step 6 On the security appliance, create a trustpoint for the Cisco Manufacturing CA and enroll via terminal by entering the following commands. Enroll via terminal because you will paste the certificate you downloaded in Step 4.

```
hostname(config)# crypto ca trustpoint trustpoint_name
hostname(config-ca-trustpoint)# enrollment terminal
```

Step 7 Authenticate the trustpoint by entering the following command:

```
hostname(config)# crypto ca authenticate trustpoint
```

Step 8 You are prompted to "Enter the base 64 encoded CA Certificate." Copy the .PEM file you downloaded in Step 4 and paste it at the command line. The file is already in base-64 encoding so no conversion is required. If the certificate is OK, you are prompted to accept it: "Do you accept this certificate? [yes/no]." Enter **yes**.

Note When you copy the certificate, make sure that you also copy also the lines with BEGIN and END.

Tip If the certificate is not ok, use the **debug crypto ca** command to show debug messages for PKI activity (used with CAs).

Step 9 Repeat the Step 1 through Step 8 for the next certificate. Table 27-5 shows the certificates that are required by the security appliance.

Table 27-5 Certificates Required by the Security Appliance for the Phone Proxy

Certificate Name	Required for...
CallManager	Authenticating the Cisco UCM during TLS handshake; only required for mixed-mode clusters.
Cisco_Manufacturing_CA	Authenticating IP phones with a Manufacturer Installed Certificate (MIC).
CAP-RTP-001	Authenticating IP phones with a MIC.
CAP-RTP-002	Authenticating IP phones with a MIC.
CAPF	Authenticating IP phones with an LSC.

Configuring the Phone Proxy in a Mixed-mode Cisco UCM Cluster

When the phone proxy is being configured to run in mixed-mode clusters, you have the following options:

- If the cluster is in mixed mode, the user has the option to use the existing CTL file to install the trustpoints.
- If a CTL file exists for the cluster, copy the CTL file to Flash memory and configure the security appliance to read from that CTL file. When you copy the CTL file to Flash memory, do not name the file `CTLFile.tlv`.

Note For mixed-mode clusters, the phone proxy does not support the Cisco Unified Call Manager using TFTP to send encrypted configuration files to IP phones through the security appliance.

Step 1 Use an existing CTL file to install the trustpoints for each entity in the network (Cisco UCM, Cisco UCM and TFTP, TFTP server, CAPF) that the IP phones must trust. If you have an existing CTL file that contains the correct IP addresses of the entities (namely, the IP address that the IP phones use for the Cisco UCM or TFTP servers), you can be use it to create a new CTL file. Store a copy of the existing CTL file to Flash memory and rename it something other than `CTLFile.tlv` and continue to Step 2.

Or

Create trustpoints and generate certificates for each entity in the network (Cisco UCM, Cisco UCM and TFTP, TFTP server, CAPF) that the IP phones must trust by performing the following substeps:

a. Create the trustpoints for each Cisco UCM (primary and secondary) by entering the following commands:

```
hostname(config)# crypto key generate rsa label keyname modulus 1024
hostname(config-ca-trustpoint)# enrollment self
hostname(config-ca-trustpoint)# keypair keyname
hostname(config-ca-trustpoint)# exit
hostname(config)# crypto ca enroll trustpoint
```

Entering these commands generates a self-signed certificate and specifies the keypair whose public key is being certified. This is the keypair created in substep a. Entering the **crypto ca enroll** command requests the certificate from the CA server and causes the security appliance to generate the certificate.

b. Create the trustpoint for the TFTP server by entering the following commands:

```
hostname(config)# crypto ca trustpoint trustpoint_name
hostname(config-ca-trustpoint)# enrollment self
hostname(config-ca-trustpoint)# keypair keyname
hostname(config-ca-trustpoint)# exit
hostname(config)# crypto ca enroll trustpoint
```

Note You are only required to perform this step when the TFTP server resides on a different server from the Cisco UCM. See Example 3: Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Different Servers, page 27-68 for an example of this configuration.

c. When prompted to include the device serial number in the subject name, type **Y** to include the serial number or type **N** to exclude it.

d. When prompted to generate the self-signed certificate, type **Y**.

e. Import the following certificates which are stored on the Cisco UCM. These certificates are required by the security appliance for the phone proxy.

- CallManager
- Cisco_Manufacturing_CA
- CAP-RTP-001
- CAP-RTP-002

See Importing Certificates from the Cisco UCM, page 27-17. For example, the CA Manufacturer certificate is required by the phone proxy to validate the IP phone certificate.

f. (Optional) If LSC provisioning is required or you have LSC enabled IP phones, you must import the CAPF certificate from the Cisco UCM. See Importing Certificates from the Cisco UCM, page 27-17.

Note If the Cisco UCM has more than one CAPF certificate, you must import all of them to the security appliance.

Step 2 Create the CTL file that will be presented to the phones during the TFTP. The *address* here must be the translated or global address of the TFTP server or Cisco UCM if NAT is configured.

a. If you are using domain names for your Cisco UCM and TFTP server, you must configure DNS lookup on the security appliance. Add an entry for each of the outside interfaces on the security appliance into your DNS server, if such entries are not already present. Each security appliance outside IP address should have a DNS entry associated with it for lookups. These DNS entries must also be enabled for Reverse Lookup. Enable DNS lookups on your security appliance with the command **dns domain-lookup** *interface_name* (where the *interface_name* specifies the interface that has a route to your DNS server). Additionally, define your DNS server IP address on the security appliance; for example: `dns name-server 10.2.3.4` (IP address of your DNS server).

Note You can enter the **dns domain-lookup** command multiple times to enable DNS lookup on multiple interfaces. If you enter multiple commands, the security appliance tries each interface in the order it appears in the configuration until it receives a response.

b. Create the CTL file instance by entering the following command:

```
hostname(config)# ctl-file ctl_name
```

c. If you are using an existing CTL file, use the trustpoints that are already in existing CTL file stored in Flash memory by entering the following command:

```
hostname(config-ctl-file)# cluster-ctl-file filename_path
```

Where the existing CTL file was saved to Flash memory with a filename other than `CTLFile.tlv`; for example, `old_ctlfile.tlv`.

Note Complete the remaining items in this step if you are creating a new CTL file instance or you want to add more entries to an existing CTL file.

d. Create the record entry for the TFTP server by entering the following command:

```
hostname(config-ctl-file)# record-entry tftp trustpoint trustpoint_name address
TFTP_IP_address
```

e. Create the record entry for the each Cisco UCM (primary and secondary) by entering the following command:

```
hostname(config-ctl-file)# record-entry cucm trustpoint trustpoint_name address
IP_address
```

f. (Optional) If LSC provisioning is required or you have LSC enabled IP phones, create the record entry for CAPF by entering the following command:

```
hostname(config-ctl-file)# record-entry capf trustpoint trustpoint_name address
IP_address
```

g. Create the CTL file by entering the following command:

```
hostname(config-ctl-file)# no shutdown
```

When the file is created, it creates an internal trustpoint used by the phone proxy to sign the TFTP files. The trustpoint is named **_internal_PP**_*ctl-instance_filename*.

h. Save the certificate configuration to Flash memory by entering the following command:

```
hostname(config)# copy running-configuration startup-configuration
```

Step 3 Create the TLS proxy instance to handle the encrypted signaling.

For mixed mode clusters, there might be IP phones that are already configured as encrypted so it requires TLS to the Cisco UCM. You must configure the LDC issuer for the TLS proxy. For more information about any of the following steps, see TLS Proxy for Encrypted Voice Inspection, page 27-42.

a. Create the necessary RSA key pairs by entering the following commands:

```
hostname(config)# crypto key generate rsa label key-pair-label modulus size
hostname(config)# crypto key generate rsa label key-pair-label modulus size
```

Where the *key-pair-label* is the LDC signer key and the key for the IP phones.

b. Create an internal local CA to sign the LDC for Cisco IP phones by entering the following commands:

```
hostname(config)# crypto ca trustpoint trustpoint_name
hostname(config-ca-trustpoint)# enrollment self
hostname(config-ca-trustpoint)# proxy-ldc-issuer
hostname(config-ca-trustpoint)# fqdn fqdn
hostname(config-ca-trustpoint)# subject-name X.500_name
hostname(config-ca-trustpoint)# keypair keypair
hostname(config)# crypto ca enroll ldc_server
```

Where the *trustpoint-name*, *fqdn*, *X.500_name*, *keypair*, and *trustpoint* are for the LDC.

c. Create the TLS proxy instance by entering the following commands:

```
hostname(config)# tls-proxy proxy_name
hostname(config-tlsp)# server trust-point _internal_PP_ctl-instance_filename
hostname(config-tlsp)# client ldc issuer ca_tp_name
hostname(config-tlsp)# client ldc keypair key_label
hostname(config-tlsp)# client cipher-suite cipher-suite
```

Where the *ca_tp_name* specifies the local CA trustpoint to issue client dynamic certificates and the *key_label* Specifies the RSA keypair to be used by client dynamic certificates.

d. Export the local CA certificate and install it as a trusted certificate on the Cisco Unified Call Manager server by performing one of the following actions:

– Use the following command to export the certificate if a trustpoint with proxy-ldc-issuer is used as the signer of the dynamic certificates:

```
hostname(config)# crypto ca export trustpoint identity-certificate
```

– For the embedded local CA server LOCAL-CA-SERVER, use the following command to export its certificate:

```
hostname(config)# show crypto ca server certificates
```

e. Save the output to a file and import the certificate on the Cisco Unified Call Manager. For more information, see the Cisco Unified Call Manager document: http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/cucos/5_0_4/iptpch6.html#wp1040848

f. Use the Display Certificates function in the Cisco Unified Call Manager software to verify the installed certificate:

http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/cucos/5_0_4/iptpch6.html#wp1040354

Step 4 Configure the phone proxy instance.

a. Create the CTL file instance:

```
hostname(config)# phone-proxy phone_proxy_name
```

b. Configure the media-termination address used by the phone-proxy for SRTP and RTP by entering the following command:

```
hostname(config-phone-proxy)# media-termination address ip_address
```

Note

- For the media termination address, you must select a publicly routable IP address that is an unused IP address on an attached outside network to the security appliance interface that will never be used by another device in your network.
- Specifically, the media termination address cannot be the same as any security appliance interface IP address, cannot overlap with existing static NAT rules, and cannot be the same as the Cisco UCM or TFTP server IP address.
- For IP phones behind a router or gateway, add routes to the media termination address on the router or gateway so that the phone can reach the media termination address.

c. Create the TFTP server using the actual internal address and specify the interface on which the TFTP server resides by entering the following command:

```
hostname(config-phone-proxy)# tftp-server address ip_address interface interface
```

d. Configure the TLS proxy instance created in Step 3 by entering the following command:

```
hostame(config-phone-proxy)# tls-proxy proxy_name
```

e. Configure the CTL file instance created in Step 2 by entering the following command:

```
hostname(config-phone-proxy)# ctl-file ctl_name
```

f. Configure the mode of the cluster to be mixed mode because the default is nonsecure.

```
hostname(config-phone-proxy)# cluster-mode mixed
```

g. (Optional) If the operational environment has an external HTTP proxy to which the IP phones direct all HTTP request, enter the following command to configure a proxy server on the security appliance:

proxy-server address *ip_address* [*listen_port*] **interface** *ifc*

You can configure only one proxy server while the phone proxy is in use; however, if the IP phones have already downloaded their configuration files after you have configured the proxy server, you must restart the IP phones so that they get the configuration file with the proxy server address in the file.

By default, the Phone URL Parameters configured under the Enterprise Parameters use an FQDN in the URLs. The parameters might need to be changed to use an IP address if the DNS lookup for the HTTP proxy does not resolve the FQDNs.

h. (Optional) To force Cisco IP Communicator (CIPC) softphones to operate in authenticated mode when CIPC softphones are deployed in a voice and data VLAN scenario, enter the following command:

```
hostname(config-phone-proxy)# cipc security-mode authenticated
```

See Phone Proxy Configuration for Cisco IP Communicator, page 27-24 for all requirements for using the phone proxy with CIPC.

i. (Optional) To preserve the settings configured on the Cisco UCM for each IP phone configured, enter the following command:

```
hostname(config-phone-proxy)# no disable service-settings
```

By default, the following settings are disabled on the IP phones:

- PC Port
- Gratuitous ARP
- Voice VLAN access
- Web Access
- Span to PC Port

Step 5 Enable the phone proxy with SIP and Skinny inspection.

a. Configure the secure Skinny class of traffic to inspect by entering the following commands:

```
hostname(config)# class-map class_map_name
hostname(config-cmap)# match port tcp eq 2443
```

b. Configure the secure SIP class of traffic to inspect by entering the following commands:

```
hostname(config)# class-map class_map_name
hostname(config-cmap)# match port tcp eq 5061
```

c. Configure the policy map and attach the action to the class of traffic by entering the following commands:

```
hostname(config)# policy-map name
hostname(config-pmap)# class classmap-name
hostname(config-pmap-c)# inspect skinny phone-proxy pp_name
hostnae(config-pmap)# class classmap-name
hostname(config-pmap-c)# inspect sip phone-proxy pp_name
```

d. Enable the policy on the outside interface by entering the following command:

```
hostname(config)# service-policy policymap_name interface intf
```

Phone Proxy Configuration for Cisco IP Communicator

To configure Cisco IP Communicator (CIPC) with the phone proxy, you must meet the following requirements:

- Include the **cipc security-mode authenticated** command under the **phone-proxy** command.
- Create an ACL to allow CIPC to register with the Cisco UCM in nonsecure mode.
- Configure null-sha1 as one of the SSL encryption ciphers.

Current versions of Cisco IP Communicator (CIPC) support authenticated mode and perform TLS signaling but not voice encryption. Therefore, you must include the following command when configuring the phone proxy instance:

cipc security-mode authenticated

Because CIPC requires an LSC to perform the TLS handshake, CIPC needs to register with the Cisco UCM in nonsecure mode using cleartext signaling. To allow the CIPC to register, create an ACL that allows the CIPC to connect to the Cisco UCM on the nonsecure SIP/SCCP signalling ports (5060/2000).

CIPC uses a different cipher when doing the TLS handshake and requires the null-sha1 cipher and SSL encryption be configured. To add the null-shal cipher, use the **show run all ssl** command to see the output for the **ssl encryption** command and add **null-shal** to the end of the SSL encryption list.

Note When used with CIPC, the phone proxy does not support end-users resetting their device name in CIPC (Prefences > Network tab > Use this Device Name field) or Administrators resetting the devide name in Cisco Unified CM Administration console (Device menu > Phone Configuration > Device Name field). To function with the phone proxy, the CIPC configuration file must be in the format: SEP<mac_address>.cnf.xml. If the device name does not follow this format (SEP<mac_address>), CIPC cannot retrieve its configuration file from Cisco UMC via the phone proxy and CIPC will not function.

The phone proxy and CIPC are not supported when CIPC is installed on computers in remote locations, such that the calls from those computers traverse the Internet, terminating at the adsecurity appliance to reach IP phones residing on the network behind the adaptive security appliance. The computers where CIPC is installed must be on the network to reach the IP phones behind the security appliance.

Configuring Linksys Routers for UDP Port Forwarding

When IP phones are behind a NAT-capable router, the router can be configured to forward the UDP ports to the IP address of the IP phone. Specifically, configure the router for UDP port forwarding when an IP phone is failing during TFTP requests and the failure is due to the router dropping incoming TFTP data packets. Configure the router to enable UDP port forwarding on port 69 to the IP phone.

As an alternative of explicit UDP forwarding, some Cable/DSL routers require you to designate the IP phone as a DMZ host. For Cable/DSL routers, this host is a special host that receives all incoming connections from the public network.

When configuring the phone proxy, there is no functional difference between an IP phone that has UDP ports explicitly forwarded or an IP phone designated as a DMZ host. The choice is entirely dependent upon the capabilities and preference of the end user.

Configuring Your Router

Your firewall/router needs to be configured to forward a range of UDP ports to the IP phone. This will allow the IP phone to receive audio when you make/receive calls.

Note Different Cable/DSL routers have different procedures for this configuration. Furthermore most NAT-capable routers will only allow a given port range to be forwarded to a single IP address

The configuration of each brand/model of firewall/router is different, but the task is the same. For specific instructions for your brand and model of router, please contact the manufacturer's website.

Linksys Routers

Step 1 From your web browser, connect to the router administrative web page. For Linksys, this is typically something like `http://192.168.1.1`.

Step 2 Click Applications & Gaming or the Port Forwarding tab (whichever is present on your router).

Step 3 Locate the table containing the port forwarding data and add an entry containing the following values:

Table 27-6 Port Forwarding Values to Add to Router

Application	Start	End	Protocol	IP Address	Enabled
IP phone	1024	65535	UDP	*Phone IP address*	**Checked**
TFTP	69	69	UDP	*Phone IP address*	**Checked**

Step 4 Click Save Settings. Port forwarding is configured.

About Rate Limiting TFTP Requests

In a remote access scenario, we recommend that you configure rate limiting of TFTP requests because any IP phone connecting through the Internet is allowed to send TFTP requests to the TFTP server.

To configure rate limiting of TFTP requests, configure the **police** command in the Modular Policy Framework. See the *Cisco Security Appliance Command Reference* for information about using the **police** command.

Policing is a way of ensuring that no traffic exceeds the maximum rate (in bits/second) that you configure, thus ensuring that no one traffic flow can take over the entire resource. When traffic exceeds the maximum rate, the security appliance drops the excess traffic. Policing also sets the largest single burst of traffic allowed.

Rate Limiting Configuration Example

The following example describes how you configure rate limiting for TFTP requests by using the **police** command and the Modular Policy Framework.

Begin by determining the conformance rate that is required for the phone proxy. To determine the conformance rate, use the following formula:

```
X * Y * 8
```

Where

X = requests per second

Y = size of each packet, which includes the L2, L3, and L4 plus the payload

Therefore, if a rate of 300 TFTP requests/second is required, then the conformance rate would be calculated as follows:

```
300 requests/second * 80 bytes * 8 = 192000
```

The example configuration below shows how the calculated conformance rate is used with the **police** command:

```
access-list tftp extended permit udp any host 192.168.0.1 eq tftp

class-map tftpclass
    match access-list tftp

policy-map tftpmap
    class tftpclass
    police output 192000

service-policy tftpmap interface inside
```

About ICMP Traffic Destined for the Media Termination Address

To control which hosts can ping the media termination address, use the **icmp** command and apply the access rule to the outside interface on the security appliance.

Any rules for ICMP access applied to the outside interface apply to traffic destined for the media termination address.

For example, use the following command to deny ICMP pings from any host destined for the media termination address:

icmp deny any outside

Troubleshooting the Phone Proxy

This section includes the following topics:

- Debugging Information from the Security Appliance, page 27-26
- Debugging Information from IP Phones, page 27-30
- IP Phone Registration Failure, page 27-31
- Media Termination Address Errors, page 27-40
- Audio Problems with IP Phones, page 27-40
- Troubleshooting the Phone Proxy, page 27-26

Debugging Information from the Security Appliance

This section describes how to use the **debug**, **capture**, and **show** commands to obtain debugging information for the phone proxy. See the *Cisco Security Appliance Command Reference* for detailed information about the syntax for these commands.

Table 27-7 lists the **debug** commands to use with the phone proxy.

Table 27-7 Security Appliance Debug Commands to Use with the Phone Proxy

To	Use the Command	Notes
To show error and event messages for TLS proxy inspection.	**debug inspect tls-proxy** [**events** | **errors**]	Use this command when your IP phone has successfully downloaded all TFTP files but is failing to complete the TLS handshake with the TLS proxy configured for the phone proxy.
To show error and event messages of media sessions for SIP and Skinny inspections related to the phone proxy.	**debug phone-proxy media** [**events** | **errors**]	Use this command in conjunction with the **debug sip** command and the **debug skinny** command if your IP phone is experiencing call failures or audio problems.
To show error and event messages of signaling sessions for SIP and Skinny inspections related to the phone proxy.	**debug phone-proxy signaling** [**events** | **errors**]	Use this command in conjunction with the **debug sip** command and the **debug skinny** command if your IP phone is failing to register with the Cisco UCM or if you are experiencing call failure.
To show error and event messages of TFTP inspection, including creation of the CTL file and configuration file parsing.	**debug phone-proxy tftp** [**events** | **errors**]	
To show debug messages for SIP application inspection.	**debug sip**	Use this command when your IP phones are experiencing connection problems; for example, you can connect within the network but cannot make calls off the network. In the output, check for 4XX or 5XX messages.
To show debug messages for SCCP (Skinny) application inspection.	**debug skinny**	Use this command when your IP phones are experiencing connection problems; for example, you can connect within the network but cannot make calls off the network. In the output, check for 4XX or 5XX messages.

Table 27-8 lists the capture commands to use with the phone proxy. Use the **capture** command on the appropriate interfaces (IP phones and Cisco UCM) to enable packet capture capabilities for packet sniffing and network fault isolation.

Table 27-8 Security Appliance Capture Commands to Use with the Phone Proxy

To	Use the Command	Notes
To capture packets on the security appliance interfaces.	**capture** *capture_name* **interface** *interface_name*	Use this command if you are experiencing any problems that might require looking into the packets. For example, if there is a TFTP failure and the output from the **debug** command does not indicate the problem clearly, run the **capture** command on the interface on which the IP phone resides and the interface on which the TFTP server resides to see the transaction and where the problem could be.
To capture data from the TLS proxy when there is a non-secure IP phone connecting to the phone proxy on the inside interface.	**capture** *capture_name* **packet-length** *bytes* **interface inside buffer** *buf_size*	
To capture encrypted data from the TLS proxy when there are secure IP phones connecting to the phone proxy on the inside interface.	**capture** *capture_name* **type tls-proxy buffer** *buf_size* **packet-length** *bytes* **interface inside**	
To capture encrypted inbound and outbound data from the TLS proxy on one or more interfaces.	**capture** *capture_name* **type tls-proxy buffer** *buf_size* **packet-length** *bytes* **interface** *interface_name*	If signaling fails, you might require capturing decrypted packets to see the contents of the SIP and SCCP signaling message. Use the **type tls-proxy** option in the **capture** command.

Table 27-9 lists the **show** commands to use with the phone proxy.

Table 27-9 Security Appliance Show Commands to Use with the Phone Proxy

To	Use the Command	Notes
To show the packets or connections dropped by the accelerated security path.	**show asp drop**	Use this command to troubleshoot audio quality issues with the IP phones or other traffic issues with the phone proxy. In addition to running this command, get call status from the phone to check for any dropped packets or jitter. See Debugging Information from IP Phones, page 27-30.
To show the classifier contents of the accelerated security path for the specific classifier domain.	**show asp table classify domain** *domain_name*	If the IP phones are not downloading TFTP files, use this command to check that the classification rule for the domain `inspect-phone-proxy` is set for hosts to the configured TFTP server under the phone proxy instance. If the IP phones are failing to register, use this command to make sure there is a classification rule for the domain `app-redirect` set for the IP phones that cannot register.
To show the connections that are to the security appliance or from the security appliance, in addition to through-traffic connections.	**show conn all**	If you are experiencing problems with audio, use this command to make sure that there are connections opened from the IP phone to the media termination address. **Note** Use the **show conn** command with following options to display TFTP connections that have replicated (unused) connections: `hostname# show conn | include p` The output for the TFTP connections should have a “p” flag at the end: `UDP out 64.169.58.181:9014 in 192.168.200.101:39420 idle 0:01:51 bytes 522 flags p` Using this command shows that the phone proxy has connections that are going through “inspect-phone-proxy”, which inspects TFTP connections. Using this command verifies that the TFTP requests are being inspected because the p flag is there.

Table 27-9 Security Appliance Show Commands to Use with the Phone Proxy

To	Use the Command	Notes
To show the logs in the buffer and logging settings.	**show logging**	Before entering the **show logging** command, enable the **logging buffered** command so that the **show logging command** displays the current message buffer and the current settings. Use this command to determine if the phone proxy and IP phones are successfully completing the TLS handshake. **Note** Using the **show logging** command is useful for troubleshooting many problems where packets might be denied or there are translation failures.
To show the corresponding media sessions stored by the phone proxy.	**show phone-proxy media-sessions**	Use this command to display output from successful calls. Additionally, use this command to troubleshoot problems with IP phone audio, such as one-way audio.
To show the IP phones capable of Secure mode stored in the database.	**show phone-proxy secure-phones**	For any problems, make sure there is an entry for the IP phone in this output and that the port for this IP phone is non-zero, which indicates that it has successfully registered with the Cisco UCM.
To show the corresponding signaling sessions stored by the phone proxy.	**show phone-proxy signaling-sessions**	Use this command to troubleshoot media or signaling failure.
To show the configured service policies.	**show service-policy**	Use this command to show statistics for the service policy.
To show active TLS proxy sessions related to the phone proxy.	**show tls-proxy sessions**	If the IP phone has failed to register, use this command to see if the IP phone has successfully completed the handshake with the TLS proxy configured for the phone proxy.

Debugging Information from IP Phones

On the IP phone, perform the following actions:

- Check the Status messages on the IP phone by selecting the **Settings** button > Status > Status Messages and selecting the status item that you want to view.
- Collect the call-statistics data from the IP phone by selecting the **Settings** button > Status > Call Statistic. Data like the following displays:

```
RxType: G.729              TxType: G.729
RxSize:   20 ms            TxSize:  20 ms
RxCnt: 0                   TxCnt: 014174
AvgJtr:   10               MaxJtr:  59
RxDisc: 0000               RxLost: 014001
```

- Check the Security settings on the IP phone by selecting the **Settings** button > Security Configuration. Settings for web access, Security mode, MIC, LSC, CTL file, trust list, and CAPF appear. Under Security mode, make sure the IP phone is set to Encrypted.
- Check the IP phone to determine which certificates are installed on the phone by selecting the **Settings** button > Security Configuration > Trust List. In the trustlist, verify the following:
  - Make sure that there is an entry for each entity that the IP phone will need to contact. If there is a primary and backup Cisco UCM, the trustlist should contain entries for each Cisco UCM.
  - If the IP phone needs an LSC, the record entry should contain a CAPF entry.
  - Make sure that the IP addresses listed for each entry are the mapped IP addresses of the entities that the IP phone can reach.
- Open a web browser and access the IP phone console logs at the URL `http://IP_phone_IP address`. The device information appears in the page. In the Device Logs section in the left pane, click Console Logs.

IP Phone Registration Failure

The following errors can make IP phones unable to register with the phone proxy:

- TFTP Auth Error Displays on IP Phone Console, page 27-31
- Configuration File Parsing Error, page 27-32
- Configuration File Parsing Error: Unable to Get DNS Response, page 27-32
- Non-configuration File Parsing Error, page 27-33
- Cisco UCM Does Not Respond to TFTP Request for Configuration File, page 27-33
- IP Phone Does Not Respond After the Security Appliance Sends TFTP Data, page 27-34
- IP Phone Requesting Unsigned File Error, page 27-35
- IP Phone Unable to Download CTL File, page 27-35
- IP Phone Registration Failure from Signaling Connections, page 27-36
- SSL Handshake Failure, page 27-38
- Certificate Validation Errors, page 27-39

TFTP Auth Error Displays on IP Phone Console

Problem The IP phone displays the following Status message:

```
TFTP Auth Error
```

Solution This Status message can indicate a problem with the IP phone CTL file.

To correct problems with the IP phone CTL file, perform the following:

Step 1 From the IP phone, select the **Setting** button > Security Configuration > Trust List. Verify that each entity in the network—Primary Cisco UCM, Secondary Cisco UCM, TFTP server—has its own entry in the trustlist and that each entity IP address is reachable by the IP phone.

Step 2 From the security appliance, verify that the CTL file for the phone proxy contains one record entry for each entity in the network—Primary Cisco UCM, Secondary Cisco UCM, TFTP server—by entering the following command:

```
hostname# show running-config all ctl-file [ctl_name]
```

Each of these record entries creates one entry on the IP phone trustlist. The phone proxy creates one entry internally with the function CUCM+TFTP.

Step 3 In the CTL file, verify that each IP address is the global or mapped IP address of the entity. If the IP phones are on multiple interfaces, additional addressing requirements apply. See Addressing Requirements for IP Phones on Multiple Interfaces, page 27-11.

Configuration File Parsing Error

Problem When the security appliance receives the configuration file from the Cisco UCM and tries to parse it, the following error appears in the debug output (**debug phone-proxy tftp errors**):

```
PP: 192.168.10.5/49357 requesting SEP00010002003.cnf.xml.sgn
PP: opened 0x193166
.......
PP: Beginning of element tag is missing, got !
PP: error parsing config file
PP: Error modifying config file, dropping packet
```

Solution Perform the following actions to troubleshoot this problem:

Step 1 Enter the following URL in a web browser to obtain the IP phone configuration file from the Cisco Unified CM Administration console:

```
http://<cucm_ip>:6970/<config_file_name>
```

For example, if the Cisco UCM IP address is 128.106.254.2 and the IP phone configuration file name is SEP000100020003.cnf.xml, enter:

```
http://128.106.254.2:6970/SEP000100020003.cnf.xml
```

Step 2 Save this file, open a case with TAC and send them this file and the output from running the **debug phone-proxy tftp** command on the security appliance.

Configuration File Parsing Error: Unable to Get DNS Response

Problem When the security appliance receives the configuration file from the Cisco UCM and tries to parse it, the following error appears in the debug output (**debug phone-proxy tftp errors**):

```
PP: 192.168.10.5/49357 requesting SEP00010002003.cnf.xml.sgn
PP: opened 0x193166
.......
PP: Callback required for parsing config file
PP: Unable to get dns response for id 7
PP: Callback, error modifying config file
```

The error indicates that the Cisco UCM is configured as an FQDN and the phone proxy is trying to do a DNS lookup but failed to get a response.

Solution

Step 1 Verify that DNS lookup is configured on the security appliance.

Step 2 If DNS lookup is configured, determine whether you can ping the FQDN for the Cisco UCM from the security appliance.

Step 3 If security appliance cannot ping the Cisco UCM FQDN, check to see if there is a problem with the DNS server.

Step 4 Additionally, use the **name** command to associate a name with an IP address with the FQDN. See the *Cisco Security Appliance Command Reference* for information about using the **name** command.

Non-configuration File Parsing Error

Problem The security appliance receives a file other than an IP phone configuration file from the Cisco UCM and attempts to parse it. The following error appears in the debug output (**debug phone-proxy tftp**):

```
PP: 192.168.10.5/49357 requesting SK72f64050-7ad5-4b47-9bfa-5e9ad9cd4aa9.xml.sgn
PP: opened 0x193166
.......
PP: Beginning of element tag is missing, got !
PP: error parsing config file
PP: Error modifying config file, dropping packet
```

Solution The phone proxy should parse only the IP phone configuration file. When the phone proxy TFTP state gets out of state, the phone proxy cannot detect when it is attempting to parse a file other than the IP phone configuration file and the error above appears in the security appliance output from the **debug phone-proxy tftp** command.

Perform the following actions to troubleshoot this problem:

Step 1 Reboot the IP phone.

Step 2 On the security appliance, enter the following command to obtain the error information from the first TFTP request to the point where the first error occurred.

```
hostname# debug phone-proxy tftp
```

Step 3 Capture the packets from the IP phone to the security appliance. Make sure to capture the packets on the interface facing the IP phone and the interface facing the Cisco UCM. See Debugging Information from the Security Appliance, page 27-26.

Step 4 Save this troubleshooting data, open a case with TAC and give them this information.

Cisco UCM Does Not Respond to TFTP Request for Configuration File

Problem When the security appliance forwards the TFTP request to the Cisco UCM for the IP phone configuration file, the Cisco UCM does not respond and the following errors appear in the debug output (**debug phone-proxy tftp**):

```
PP: 192.168.10.5/49355 requesting SEP001562106AF3.cnf.xml.sgn
PP: opened 0x17ccde
PP: 192.168.10.5/49355 requesting SEP001562106AF3.cnf.xml.sgn
PP: Client outside:192.168.10.5/49355 retransmitting request for Config file
SEP001562106AF3.cnf.xml.sgn
PP: opened 0x17ccde
PP: 192.168.10.5/49355 requesting SEP001562106AF3.cnf.xml.sgn
```

```
PP: Client outside:192.168.10.5/49355 retransmitting request for Config file
SEP001562106AF3.cnf.xml.sgn
PP: opened 0x17ccde
PP: 192.168.10.5/49355 requesting SEP001562106AF3.cnf.xml.sgn
PP: Client outside:192.168.10.5/49355 retransmitting request for Config file
SEP001562106AF3.cnf.xml.sgn
PP: opened 0x17ccde
```

Solution Perform the following actions to troubleshoot this problem:

Step 1 Determine why the Cisco UCM is not responding to the TFTP request by performing the following troubleshooting actions:

- Use the Cisco UCM to ping the security appliance inside interface when PAT is configured for the outside interface so that the IP phone IP address is uses NAT for the security appliance inside interface IP address.
- Use the Cisco UCM to ping the IP phone IP address when NAT and PAT are not configured.

Step 2 Verify that the security appliance is forwarding the TFTP request. Capture the packets on the interface between the security appliance and Cisco UCM. See Debugging Information from the Security Appliance, page 27-26.

IP Phone Does Not Respond After the Security Appliance Sends TFTP Data

Problem When the security appliance receives a TFTP request from the IP phone for the CTL file and forwards the data to the IP phone, the phone might not see the data and the TFTP transaction fails.

The following errors appear in the debug output (**debug phone-proxy tftp**):

```
PP: Client outside:68.207.118.9/33606 retransmitting request for CTL file
CTLSEP001DA2B78E91.tlv
PP: opened 0x214b27a
PP: Data Block 1 forwarded from 168.215.146.220/20168 to 68.207.118.9/33606 ingress ifc
outside
PP: 68.207.118.9/33606 requesting CTLSEP001DA2B78E91.tlv
PP: Client outside:68.207.118.9/33606 retransmitting request for CTL file
CTLSEP001DA2B78E91.tlv
PP: 68.207.118.9/33606 requesting CTLSEP001DA2B78E91.tlv
PP: Client outside:68.207.118.9/33606 retransmitting request for CTL file
CTLSEP001DA2B78E91.tlv
```

Solution Perform the following actions to determine why the IP phone is not responding and to troubleshoot the problem:

Step 1 Verify that the security appliance is forwarding the TFTP request by entering the following command to capture the packets on the interface between the security appliance and the IP phone:

```
hostname# capture out interface outside
```

See the *Cisco Security Appliance Command Reference* for more information about using the **capture** command.

Step 2 If the IP phone is behind a router, the router might be dropping the data. Make sure UDP port forwarding is enabled on the router.

Step 3 If the router is a Linksys router, see Configuring Linksys Routers for UDP Port Forwarding, page 27-24 for information on the configuration requirements.

IP Phone Requesting Unsigned File Error

Problem The IP phone should always request a signed file. Therefore, the TFTP file being requested always has the .SGN extension.

When the IP phone does not request a signed file, the following error appears in the debug output (**debug phone-proxy tftp errors**):

```
Error: phone requesting for unsigned config file
```

Solution Most likely, this error occurs because the IP phone has not successfully installed the CTL file from the security appliance.

Determine whether the IP phone has successfully downloaded and installed the CTL file from the security appliance by checking the Status messages on the IP phone. See Debugging Information from IP Phones, page 27-30 for information.

IP Phone Unable to Download CTL File

Problem The IP phone Status message indicates it cannot download its CTL file and the IP phone cannot be converted to Secure (encrypted) mode.

Solution If the IP phone did not have an existing CTL file, check the Status messages by selecting the **Settings** button > Status > Status Messages. If the list contains a Status message indicating the IP phone encountered a CTL File Auth error, obtain the IP phone console logs, open a TAC case, and send them the logs.

Solution This error can appear in the IP phone Status messages when the IP phone already has an existing CTL file.

Step 1 Check the IP phone to see if a CTL file already exists on it. This can occur if the IP phone previously registered with a mixed mode cluster Cisco UCM. On the IP phone, select the **Settings** button > Security Configuration > CTL file.

Step 2 Erase the existing CTL file by selecting the **Settings** button > Security Configuration > CTL file > Select. Press **# on the keypad and select Erase.

Solution Problems downloading the CTL file might be caused by issues with media termination. Enter the following command to determine if the media-termination address in the phone proxy configuration is set correctly:

```
hostname(config)# show running-config all phone-proxy
!
phone-proxy mypp
    media-termination address 10.10.0.25
    cipc security-mode authenticated
    cluster-mode mixed
    disable service-settings
    timeout secure-phones 0:05:00
hostname(config)#
```

Make sure that the media-termination address is set correctly. The security appliance must have an IP address for media termination that meets the following criteria:

- The IP address is a publicly routable address that is an unused IP address on an attached outside network to the security appliance interface that will never be used by another device in your network.
- The IP address cannot be the same as any of the security appliance interface IP addresses.
- The IP address cannot overlap with existing static NAT rules.
- The IP address cannot be the same as the Cisco UCM or TFTP server IP address.
- For IP phones behind a router or gateway, add routes to the media termination address on the router or gateway so that the phone can reach the media termination address.

IP Phone Registration Failure from Signaling Connections

Problem The IP phone is unable to complete the TLS handshake with the phone proxy and download its files using TFTP.

Solution

Step 1 Determine if the TLS handshake is occurring between the phone proxy and the IP phone, perform the following:

a. Enable logging with the following command:

```
hostname(config)# logging buffered debugging
```

b. To check the output from the syslogs captured by the **logging buffered** command, enter the following command:

```
hostname# show logging
```

The syslogs will contain information showing when the IP phone is attempting the TLS handshake, which happens after the IP phone downloads its configuration file.

Step 2 Determine if the TLS proxy is configured correctly for the phone proxy:

a. Display all currently running TLS proxy configurations by entering the following command:

```
hostname# show running-config tls-proxy
    tls-proxy proxy
    server trust-point _internal_PP_<ctl_file_instance_name>
    client ldc issuer ldc_signer
    client ldc key-pair phone_common
    no client cipher-suite
hostname#
```

b. Verify that the output contains the **server trust-point** command under the **tls-proxy** command (as shown in substep a.).

If you are missing the **server trust-point** command, modify the TLS proxy in the phone proxy configuration.

See Step 3 in Configuring the Phone Proxy in a Non-secure Cisco UCM Cluster, page 27-13, or Step 3 in Configuring the Phone Proxy in a Mixed-mode Cisco UCM Cluster, page 27-19.

Having this command missing from the TLS proxy configuration for the phone proxy will cause TLS handshake failure.

Step 3 Verify that all required certificates are imported into the security appliance so that the TLS handshake will succeed.

a. Determine which certificates are installed on the security appliance by entering the following command:

```
hostname# show running-config crypto
```

Additionally, determine which certificates are installed on the IP phones. See Debugging Information from IP Phones, page 27-30 for information about checking the IP phone to determine if it has MIC installed on it.

b. Verify that the list of installed certificates contains all required certificates for the phone proxy.

See Table 27-5, Certificates Required by the Security Appliance for the Phone Proxy, for information.

c. Import any missing certificates onto the security appliance. See also Importing Certificates from the Cisco UCM, page 27-17.

Step 4 If the steps above fail to resolve the issue, perform the following actions to obtain additional troubleshooting information for Cisco Support.

a. Enter the following commands to capture additional debugging information for the phone proxy:

```
hostname# debug inspect tls-proxy error
hostname# show running-config ssl
hostname(config) show tls-proxy tls_name session host host_addr detail
```

b. Enable the **capture** command on the inside and outside interfaces (IP phones and Cisco UCM) to enable packet capture capabilities for packet sniffing and network fault isolation. See the *Cisco Security Appliance Command Reference* for information.

Problem The TLS handshake succeeds, but signaling connections are failing.

Solution Perform the following actions:

- Check to see if SIP and Skinny signaling is successful by using the following commands:
  - **debug sip**
  - **debug skinny**
- If the TLS handshake is failing and you receive the following syslog, the SSL encryption method might not be set correctly:

```
%ASA-6-725001: Starting SSL handshake with client dmz:171.169.0.2/53097 for TLSv1
session.
%ASA-7-725010: Device supports the following 1 cipher(s).
%ASA-7-725011: Cipher[1] : RC4-SHA
%ASA-7-725008: SSL client dmz:171.169.0.2/53097 proposes the following 2 cipher(s).
%ASA-7-725011: Cipher[1] : AES256-SHA
%ASA-7-725011: Cipher[2] : AES128-SHA
%ASA-7-725014: SSL lib error. Function: SSL3_GET_CLIENT_HELLO Reason: no shared cipher
%ASA-6-725006: Device failed SSL handshake with dmz client:171.169.0.2/53097
```

Set the correct ciphers by completing the following procedure:

Step 1 To see the ciphers being used by the phone proxy, enter the following command:

```
hostname# show run all ssl
```

Step 2 To add the required ciphers, enter the following command:

```
hostname(config)# ssl encryption
```

The default is to have all algorithms available in the following order:

[3des-sha1] [des-sha1] [rc4-md5] [possibly others]

See the *Cisco Security Appliance Command Reference* for more information about setting ciphers with the **ssl encryption** command.

SSL Handshake Failure

Problem The phone proxy is not functioning. Initial troubleshooting uncovered the following errors in the security appliance syslogs:

```
%ASA-7-725014: SSL lib error. Function: SSL3_READ_BYTES Reason: ssl handshake failure
%ASA-7-725014: SSL lib error. Function: SSL3_GET_CLIENT_CERTIFICATE Reason: no certificate
returned
%ASA-6-725006: Device failed SSL handshake with outside client:72.146.123.158/30519
%ASA-3-717009: Certificate validation failed. No suitable trustpoints found to validate
certificate serial number: 62D06172000000143FCC, subject name:
cn=CP-7962G-SEP002155554502,ou=EVVBU,o=Cisco Systems Inc.
%ASA-3-717027: Certificate chain failed validation. No suitable trustpoint was found to
validate chain.
```

Solution

Verify that all required certificates are imported into the security appliance so that the TLS handshake will succeed.

Step 1 Determine which certificates are installed on the security appliance by entering the following command:

```
hostname# show running-config crypto
```

Additionally, determine which certificates are installed on the IP phones. See Debugging Information from IP Phones, page 27-30 for information about checking the IP phone to determine if it has MIC installed on it.

Step 2 Verify that the list of installed certificates contains all required certificates for the phone proxy.

See Table 27-5, Certificates Required by the Security Appliance for the Phone Proxy, for information.

Step 3 Import any missing certificates onto the security appliance. See also Importing Certificates from the Cisco UCM, page 27-17.

Problem The phone proxy is not functioning. Initial troubleshooting uncovered the following errors in the security appliance syslogs:

```
%ASA-6-725001: Starting SSL handshake with client dmz:171.169.0.2/53097 for TLSv1
session.
%ASA-7-725010: Device supports the following 1 cipher(s).
%ASA-7-725011: Cipher[1] : RC4-SHA
%ASA-7-725008: SSL client dmz:171.169.0.2/53097 proposes the following 2 cipher(s).
%ASA-7-725011: Cipher[1] : AES256-SHA
%ASA-7-725011: Cipher[2] : AES128-SHA
%ASA-7-725014: SSL lib error. Function: SSL3_GET_CLIENT_HELLO Reason: no shared cipher
%ASA-6-725006: Device failed SSL handshake with dmz client:171.169.0.2/53097
```

Solution the SSL encryption method might not be set correctly. Set the correct ciphers by completing the following procedure:

Step 1 To see the ciphers being used by the phone proxy, enter the following command:

```
hostname# show run all ssl
```

Step 2 To add the required ciphers, enter the following command:

```
hostname(config)# ssl encryption
```

The default is to have all algorithms available in the following order:

[3des-sha1] [des-sha1] [rc4-md5] [possibly others]

See the *Cisco Security Appliance Command Reference* for more information about setting ciphers with the **ssl encryption** command.

Certificate Validation Errors

Problem Errors in the security appliance log indicate that certificate validation errors occurred.

Entering the **show logging asdm** command, displayed the following errors:

```
3|Jun 19 2008 17:23:54|717009: Certificate validation failed. No suitable trustpoints
found to validate
certificate serial number: 348FD2760000000E6E27, subject name:
cn=CP-7961G-SEP001819A89CC3,ou=EVVBU,o=Cisco Systems Inc.
```

Solution

In order for the phone proxy to authenticate the MIC provided by the IP phone, it needs the Cisco Manufacturing CA (MIC) certificate imported into the security appliance.

Verify that all required certificates are imported into the security appliance so that the TLS handshake will succeed.

Step 1 Determine which certificates are installed on the security appliance by entering the following command:

```
hostname# show running-config crypto
```

Additionally, determine which certificates are installed on the IP phones. The certificate information is shown under the Security Configuration menu. See Debugging Information from IP Phones, page 27-30 for information about checking the IP phone to determine if it has the MIC installed on it.

Step 2 Verify that the list of installed certificates contains all required certificates for the phone proxy.

See Table 27-5, Certificates Required by the Security Appliance for the Phone Proxy, for information.

Step 3 Import any missing certificates onto the security appliance. See also Importing Certificates from the Cisco UCM, page 27-17.

Media Termination Address Errors

Problem Entering the **media-termination address** command displays the following errors:

```
hostname(config-phone-proxy)# media-termination address ip_address
ERROR: Failed to apply IP address to interface Virtual254, as the network overlaps with
interface GigabitEthernet0/0. Two interfaces cannot be in the same subnet.
ERROR: Failed to set IP address for the Virtual interface
ERROR: Could not bring up Phone proxy media termination interface
ERROR: Failed to find the HWIDB for the Virtual interface
```

Solution Enter the following command to determine if the media-termination address in the phone proxy configuration is set correctly:

```
hostname(config)# show running-config all phone-proxy
asa2(config)# show running-config all phone-proxy
!
phone-proxy mypp
    media-termination address 10.10.0.25
    cipc security-mode authenticated
    cluster-mode mixed
    disable service-settings
    timeout secure-phones 0:05:00
hostname(config)#
```

Make sure that the media-termination address is set correctly. The security appliance must have an IP address for media termination that meets the following criteria:

- The IP address is a publicly routable address that is an unused IP address on an attached outside network to the security appliance interface that will never be used by another device in your network.
- The IP address cannot be the same as any of the security appliance interface IP addresses.
- The IP address cannot overlap with existing static NAT rules.
- The IP address cannot be the same as the Cisco UCM or TFTP server IP address.
- For IP phones behind a router or gateway, add routes to the media termination address on the router or gateway so that the phone can reach the media termination address.

Audio Problems with IP Phones

The following audio errors can occur when the IP phones connecting through the phone proxy.

Media Failure for a Voice Call

Problem The call signaling completes but there is one way audio or no audio.

Solution

- Problems with one way or no audio might be caused by issues with media termination. Enter the following command to determine if the media-termination address in the phone proxy configuration is set correctly:

  ```
  hostname(config)# show running-config all phone-proxy
  asa2(config)# show running-config all phone-proxy
  !
  phone-proxy mypp
      media-termination address 10.10.0.25
      cipc security-mode authenticated
      cluster-mode mixed
  ```

```
    disable service-settings
    timeout secure-phones 0:05:00
hostname(config)#
```

Make sure that the media-termination address is set correctly. The security appliance must have an IP address for media termination that meets the following criteria:

- The IP address is a publicly routable address that is an unused IP address on an attached outside network to the security appliance interface that will never be used by another device in your network.
- The IP address cannot be the same as any of the security appliance interface IP addresses.
- The IP address cannot overlap with existing static NAT rules.
- The IP address cannot be the same as the Cisco UCM or TFTP server IP address.
- For IP phones behind a router or gateway, add routes to the media termination address on the router or gateway so that the phone can reach the media termination address.

- If the media-termination address meets the requirements, determine whether the IP address is reachable by all IP phones.
- If the IP address is set correctly and it is reachable by all IP phones, check the call statistics on an IP phone (see Debugging Information from IP Phones, page 27-30) and determine if there are Rcvr packets and Sender packets on the IP phone, or if there are any Rcvr Lost or Discarded packets.

Saving SAST Keys

Site Administrator Security Token (SAST) keys on the security appliance can be saved in the event a recovery is required due to hardware failure and a replacement is required. The following steps shows how to recover the SAST keys and use them on the new hardware.

The SAST keys can be seen via the **show crypto key mypubkey rsa** command. The SAST keys are associated with a trustpoint that is labeled **_internal**_*ctl-file_name*_**SAST**_*X* where *ctl-file-name* is the name of the CTL file instance that was configured, and *X* is an integer from 0 to N-1 where N is the number of SASTs configured for the CTL file (the default is 2).

Step 1 On the security appliance, export all the SAST keys in PKCS-12 format by using the **crypto ca export** command:

```
hostname(config)# crypto ca export _internal_ctl-file_name_SAST_X pkcs12 passphrase

hostname(config)# Exported pkcs12 follows:
MIIGZwIBAzCCBiEGCSqGSIb3DQEHAaCCBhIEggYOMIIGCjCCBgYGCSqGSIb3DQEH

[snip]

MIIGZwIBAzCCBiEGCSqGSIb3DQEHAaCCBhIEggYOMIIGCjCCBgYGCSqGSIb3DQEH
---End - This line not part of the pkcs12---

hostname(config)# crypto ca export _internal_ctl-file_name_SAST_X pkcs12 passphrase

hostname(config)# Exported pkcs12 follows:
MIIGZwIBAzCCBiEGCSqGSIb3DQEHAaCCBhIEggYOMIIGCjCCBgYGCSqGSIb3DQEH

[snip]

mGF/hfDDNAICBAA=

---End - This line not part of the pkcs12---
```

```
hostname(config)#
```

Note Save this output somewhere secure.

Step 2 Import the SAST keys to a new security appliance.

a. To import the SAST key, enter the following command:

```
hostname(config)# crypto ca import trustpoint pkcs12 passphrase
```

Where *trustpoint* is **_internal**_*ctl-file_name*_**SAST**_*X* and *ctl-file-name* is the name of the CTL file instance that was configured, and *X* is an integer from 0 to 4 depending on what you exported from the security appliance.

b. Using the PKCS-12 output you saved in Step 1, enter the following command and paste the output when prompted:

```
hostname(config)# crypto ca import _internal_ctl-file_name_SAST_X pkcs12 passphrase

hostname(config)# Enter the base 64 encoded pkcs12.
hostname(config)# End with the word "quit" on a line by itself:
MIIGZwIBAzCCBiEGCSqGSIb3DQEHAaCCBhIEggYOMIIGCjCCBgYGCSqGSIb3DQEH

[snip]

muMiZ6eClQICBAA=
hostname(config)# quit
INFO: Import PKCS12 operation completed successfully
hostname(config)# crypto ca import _internal_ctl-file_name_SAST_X pkcs12 passphrase

hostname(config)# Enter the base 64 encoded pkcs12.
hostname(config)# End with the word "quit" on a line by itself:
MIIGZwIBAzCCBiEGCSqGSIb3DQEHAaCCBhIEggYOMIIGCjCCBgYGCSqGSIb3DQEH

[snip]

mGF/hfDDNAICBAA=
hostname(config)# quit
INFO: Import PKCS12 operation completed successfully
hostname(config)#
```

Step 3 Create the CTL file instance on the new security appliance using the same name as the one used in the SAST trustpoints created in Step 2 by entering the following commands. Create trustpoints for each Cisco UMC (primary and secondary).

```
hostname(config)# ctl-file ctl_name
hostname(config-ctl-file)# record-entry cucm trustpoint trust_point address address
hostname(config-ctl-file)# record-entry capf trustpoint trust_point address address
hostname(config-ctl-file)# no shutdown
```

TLS Proxy for Encrypted Voice Inspection

This section describes TLS proxy for encrypted voice inspection. This section includes the following topics:

- Overview, page 27-43
- Configuring TLS Proxy, page 27-43

- Debugging TLS Proxy, page 27-47
- CTL Client, page 27-50

Overview

With encrypted voice inspection, the security appliance decrypts, inspects and modifies (as needed, for example, performing NAT fixup), and re-encrypts voice signaling traffic while all of the existing VoIP inspection functions for Skinny and SIP protocols are preserved. Once voice signaling is decrypted, the plaintext signaling message is passed to the existing inspection engines.

The security appliance acts as a TLS proxy between the Cisco IP Phone and Cisco Unified CallManager. The proxy is transparent for the voice calls between the phone and the Cisco Unified CallManager. Cisco IP Phones download a Certificate Trust List from the Cisco Unified CallManager before registration which contains identities (certificates) of the devices that the phone should trust, such as TFTP servers and Cisco Unified CallManager servers. To support server proxy, the CTL file must contain the certificate that the security appliance creates for the Cisco Unified CallManagers. To proxy calls on behalf of the Cisco IP Phone, the security appliance presents a certificate that the Cisco Unified CallManager can verify, which is a Local Dynamic Certificate for the phone, issued by the certificate authority on the security appliance.

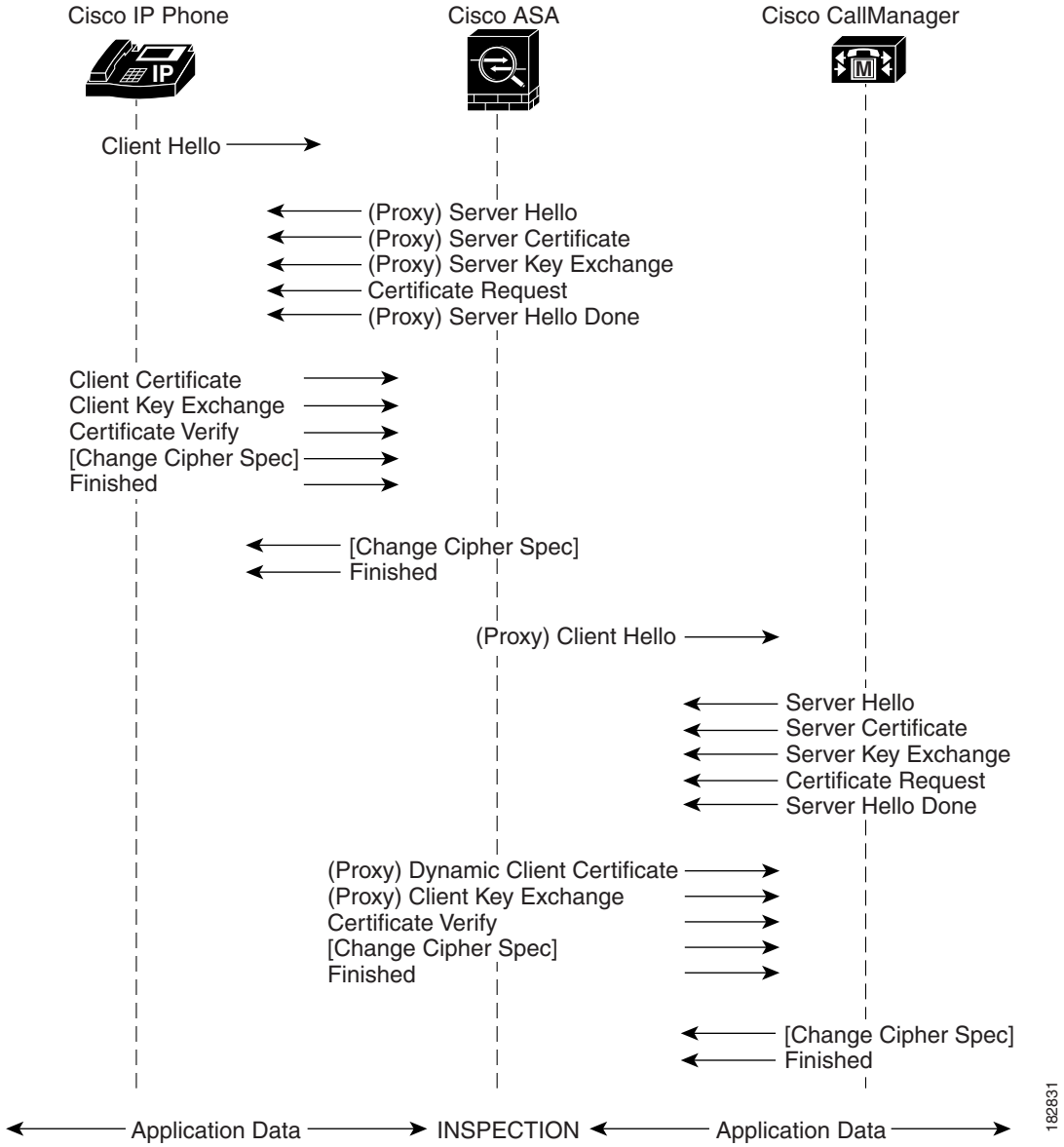
TLS proxy is supported by the Cisco Unified CallManager Release 5.1 and later. You should be familiar with the security features of the Cisco Unified CallManager. For background and detailed description of Cisco Unified CallManager security, see the Cisco Unified CallManager document.

TLS proxy applies to the encryption layer and must be configured with an application layer protocol inspection. You should be familiar with the inspection features on the ASA security appliance, especially Skinny and SIP inspection. For more information on deployment topologies and configuration, refer to the *Cisco Security Appliance Command Line Configuration Guide*.

Configuring TLS Proxy

The security appliance in Figure 27-2 serves as a proxy for both client and server, with Cisco IP Phone and Cisco Unified CallManager interaction.

Figure 27-2 TLS Proxy Flow



Before configuring TLS proxy, the following prerequisites are required:

- You must set clock on the security appliance before configuring TLS proxy. To set the clock manually and display clock, use the **clock set** and **show clock** commands. We recommend that the security appliance use the same NTP server as the Cisco Unified CallManager cluster. TLS handshake may fail due to certificate validation failure if clock is out of sync between the security appliance and the Cisco Unified CallManager server.
- 3DES-AES license is needed to interoperate with the Cisco Unified CallManager. AES is the default cipher used by the Cisco Unified CallManager and Cisco IP Phone.

To configure the security appliance for TLS proxy, perform the following steps:

Step 1 (Optional) Set the maximum number of TLS proxy sessions to be supported by the security appliance using the following command, for example:

```
hostname(config)# tls-proxy maximum-sessions 1200
```

Note The **tls-proxy maximum-sessions** command controls the memory size reserved for cryptographic applications such as TLS proxy. Crypto memory is reserved at the time of system boot. You may need to reboot the security appliance for the configuration to take effect if the configured maximum sessions number is greater than the currently reserved.

Step 2 Import the following certificates which are stored on the Cisco UCM. These certificates are required by the security appliance for the phone proxy. For example, the CA Manufacturer certificate is required by the phone proxy to validate the IP phone certificate. See Importing Certificates from the Cisco UCM, page 27-17.

- CallManager
- Cisco_Manufacturing_CA
- CAP-RTP-001
- CAP-RTP-002
- (Optional) CAPF—import the CAPF certificate when LSC provisioning is required or you have LSC enabled IP phones

Note If the Cisco UCM has more than one CAPF certificate, you must import all of them to the security appliance.

Step 3 Create necessary RSA key pairs using the following commands, for example:

```
hostname(config)# crypto key generate rsa label ccm_proxy_key modulus 1024
hostname(config)# crypto key generate rsa label ldc_signer_key modulus 1024
hostname(config)# crypto key generate rsa label phone_common modulus 1024
```

We recommend to use a different key pair for each role.

Step 4 Create the proxy certificate for the Cisco Unified CallManager cluster using the following commands, for example:

```
hostname(config)# ! for self-signed CCM proxy certificate
hostname(config)# crypto ca trustpoint ccm_proxy
hostname(config-ca-trustpoint)# enrollment self
hostname(config-ca-trustpoint)# fqdn none
hostname(config-ca-trustpoint)# subject-name cn=EJW-SV-1-Proxy
hostname(config-ca-trustpoint)# keypair ccm_proxy_key
hostname(config)# crypto ca enroll ccm_proxy
```

The Cisco Unified CallManager proxy certificate could be self-signed or issued by a third-party CA. The certificate is exported to the CTL client.

Note Cisco IP Phones require certain fields from the X.509v3 certificate to be present to validate the certificate via consulting the CTL file. Consequently, the **subject-name** entry must be configured for a proxy certificate trustpoint. The subject name must be composed of the ordered concatenation of the CN, OU and O fields. The CN field is mandatory; the others are optional.

Each of the concatenated fields (when present) are separated by a semicolon, yielding one of the

following forms:

CN=xxx;OU=yyy;O=zzz
CN=xxx;OU=yyy
CN=xxx;O=zzz
CN=xxx

Step 5 Create an internal local CA to sign the LDC for Cisco IP Phones using the following commands, for example:

```
hostname(config)# ! for the internal local LDC issuer
hostname(config)# crypto ca trustpoint ldc-server
hostname(config-ca-trustpoint)# enrollment self
hostname(config-ca-trustpoint)# proxy-ldc-issuer
hostname(config-ca-trustpoint)# fqdn my-ldc-ca.exmaple.com
hostname(config-ca-trustpoint)# subject-name cn=FW-LDC-SIGNER-172_23_45_200
hostname(config-ca-trustpoint)# keypair ldc-signer-key
hostname(config)# crypto ca enroll ldc-server
```

This local CA is created as a regular self-signed trustpoint with **proxy-ldc-issuer** enabled. You may use the embedded local CA LOCAL-CA-SERVER on the security appliance to issue the LDC.

Step 6 Create a CTL Provider instance in preparation for a connection from the CTL Client using the following commands, for example:

```
hostname(config)# ctl-provider my_ctl
hostname(config-ctl-provider)# client interface inside address 172.23.45.1
hostname(config-ctl-provider)# client username CCMAdministrator password XXXXXX encrypted
hostname(config-ctl-provider)# export certificate ccm_proxy
hostname(config-ctl-provider)# ctl install
```

The username and password must match the username and password for Cisco Unified CallManager administration. The trustpoint name in the **export** command is the proxy certificate for the Cisco Unified CallManager server.

The default port number listened by the CTL Provider is TCP 2444, which is the default CTL port on the Cisco Unified CallManager. Use the **service port** command to change the port number if a different port is used by the Cisco Unified CallManager cluster.

Step 7 Create a TLS proxy instance using the following commands, for example:

```
hostname(config)# tls-proxy my_proxy
hostname(config-tlsp)# server trust-point ccm_proxy
hostname(config-tlsp)# client ldc issuer ldc_server
hostname(config-tlsp)# client ldc key-pair phone_common
hostname(config-tlsp)# client cipher-suite aes128-sha1 aes256-sha1
```

The **server** commands configure the proxy parameters for the original TLS server. In other words, the parameters for the security appliance to act as the server during a TLS handshake, or facing the original TLS client. The **client** commands configure the proxy parameters for the original TLS client. In other words, the parameters for the security appliance to act as the client during a TLS handshake, or facing the original TLS server.

Step 8 Enable TLS proxy for the Cisco IP Phones and Cisco Unified CallManagers in Skinny or SIP inspection using the following commands, for example:

```
hostname(config)# class-map sec_skinny
hostname(config-cmap)# match port tcp eq 2443

hostname(config)# policy-map type inspect skinny skinny_inspect
hostname(config-pmap)# parameters
hostname(config-pmap-p)# ! Skinny inspection parameters
```

```
hostname(config)# policy-map global_policy
hostname(config-pmap)# class inspection_default
hostname(config-pmap-c)# inspect skinny skinny_inspect
hostname(config-pmap)# class sec_skinny
hostname(config-pmap-c)# inspect skinny skinny_inspect tls-proxy my_proxy

hostname(config)# service-policy global_policy global
```

Step 9 Export the local CA certificate (ldc_server) and install it as a trusted certificate on the Cisco Unified CallManager server.

a. Use the following command to export the certificate if a trust-point with **proxy-ldc-issuer** is used as the signer of the dynamic certificates, for example:

```
hostname(config)# crypto ca export ldc_server identity-certificate
```

b. For the embedded local CA server LOCAL-CA-SERVER, use the following command to export its certificate, for example:

```
hostname(config)# show crypto ca server certificate
```

Save the output to a file and import the certificate on the Cisco Unified CallManager. For more information, see the Cisco Unified CallManager document:
http://www.cisco.com/univercd/cc/td/doc/product/voice/c_callmg/5_0/iptp_adm/504/iptpch6.htm#wp1040848

After this step, you may use the Display Certificates function on the Cisco Unified CallManager GUI to verify the installed certificate:

http://www.cisco.com/univercd/cc/td/doc/product/voice/c_callmg/5_0/iptp_adm/504/iptpch6.htm#wp1040354

Step 10 Run the CTL Client application to add the server proxy certificate (ccm_proxy) to the CTL file and install the CTL file on the security appliance. See the Cisco Unified CallManager document for information on how to configure and use CTL Client:

http://www.cisco.com/univercd/cc/td/doc/product/voice/c_callmg/5_1/nci/p08/secuauth.htm

Note You will need the CTL Client that is released with Cisco Unified CallManager Release 5.1 to interoperate with the security appliance. See the "CTL Client" section on page 27-50 for more information regarding TLS proxy support.

Debugging TLS Proxy

You may enable TLS proxy debug flags along with SSL syslogs to debug TLS proxy connection problems. For example, using the following commands to enable TLS proxy-related debug and syslog output only:

```
hostname(config)# debug inspect tls-proxy events
hostname(config)# debug inspect tls-proxy errors
hostname(config)# logging enable
hostname(config)# logging timestamp
hostname(config)# logging list loglist message 711001
hostname(config)# logging list loglist message 725001-725014
hostname(config)# logging list loglist message 717001-717038
hostname(config)# logging buffer-size 1000000
```

```
hostname(config)# logging buffered loglist
hostname(config)# logging debug-trace
```

The following is sample output reflecting a successful TLS proxy session setup for a SIP phone:

```
hostname(config)# show log

Apr 17 2007 23:13:47: %ASA-6-725001: Starting SSL handshake with client
outside:133.9.0.218/49159 for TLSv1 session.
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Set up proxy for Client
outside:133.9.0.218/49159 <-> Server inside:195.168.2.201/5061
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Using trust point 'local_ccm' with the
Client, RT proxy cbae1538
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Waiting for SSL handshake from Client
outside:133.9.0.218/49159.
Apr 17 2007 23:13:47: %ASA-7-725010: Device supports the following 4 cipher(s).
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[1] : RC4-SHA
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[2] : AES128-SHA
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[3] : AES256-SHA
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[4] : DES-CBC3-SHA
Apr 17 2007 23:13:47: %ASA-7-725008: SSL client outside:133.9.0.218/49159 proposes the
following 2 cipher(s).
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[1] : AES256-SHA
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[2] : AES128-SHA
Apr 17 2007 23:13:47: %ASA-7-725012: Device chooses cipher : AES128-SHA for the SSL
session with client outside:133.9.0.218/49159
Apr 17 2007 23:13:47: %ASA-7-725014: SSL lib error. Function: SSL23_READ Reason: ssl
handshake failure
Apr 17 2007 23:13:47: %ASA-7-717025: Validating certificate chain containing 1
certificate(s).
Apr 17 2007 23:13:47: %ASA-7-717029: Identified client certificate within certificate
chain. serial number: 01, subject name: cn=SEP0017593F50A8.
Apr 17 2007 23:13:47: %ASA-7-717030: Found a suitable trustpoint
_internal_ejw-sv-2_cn=CAPF-08a91c01 to validate certificate.
Apr 17 2007 23:13:47: %ASA-6-717022: Certificate was successfully validated. serial
number: 01, subject name:  cn=SEP0017593F50A8.
Apr 17 2007 23:13:47: %ASA-6-717028: Certificate chain was successfully validated with
warning, revocation status was not checked.
Apr 17 2007 23:13:47: %ASA-6-725002: Device completed SSL handshake with client
outside:133.9.0.218/49159
Apr 17 2007 23:13:47: %ASA-6-725001: Starting SSL handshake with server
inside:195.168.2.201/5061 for TLSv1 session.
Apr 17 2007 23:13:47: %ASA-7-725009: Device proposes the following 2 cipher(s) to server
inside:195.168.2.201/5061
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[1] : AES128-SHA
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[2] : AES256-SHA
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Generating LDC for client
'cn=SEP0017593F50A8', key-pair 'phone_common', issuer 'LOCAL-CA-SERVER', RT proxy cbae1538
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Started SSL handshake with Server
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Data channel ready for the Client
Apr 17 2007 23:13:47: %ASA-7-725013: SSL Server inside:195.168.2.201/5061 choose cipher :
AES128-SHA
Apr 17 2007 23:13:47: %ASA-7-717025: Validating certificate chain containing 1
certificate(s).
Apr 17 2007 23:13:47: %ASA-7-717029: Identified client certificate within certificate
chain. serial number: 76022D3D9314743A, subject name: cn=EJW-SV-2.inside.com.
Apr 17 2007 23:13:47: %ASA-6-717022: Certificate was successfully validated. Certificate
is resident and trusted, serial number: 76022D3D9314743A, subject name:
cn=EJW-SV-2.inside.com.
Apr 17 2007 23:13:47: %ASA-6-717028: Certificate chain was successfully validated with
revocation status check.
Apr 17 2007 23:13:47: %ASA-6-725002: Device completed SSL handshake with server
inside:195.168.2.201/5061
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Data channel ready for the Server
```

Use the **show tls-proxy** commands with different options to check the active TLS proxy sessions. The following are some sample outputs:

```
hostname(config-tlsp)# show tls-proxy
Maximum number of sessions: 1200

TLS-Proxy 'sip_proxy': ref_cnt 1, seq# 3
    Server proxy:
        Trust-point: local_ccm
    Client proxy:
        Local dynamic certificate issuer: LOCAL-CA-SERVER
        Local dynamic certificate key-pair: phone_common
        Cipher suite:  aes128-sha1 aes256-sha1
    Run-time proxies:
         Proxy 0xcbae1538: Class-map: sip_ssl, Inspect: sip
             Active sess 1, most sess 3, byte 3456043

TLS-Proxy 'proxy': ref_cnt 1, seq# 1
    Server proxy:
        Trust-point: local_ccm
    Client proxy:
        Local dynamic certificate issuer: ldc_signer
        Local dynamic certificate key-pair: phone_common
        Cipher-suite: <unconfigured>
    Run-time proxies:
        Proxy 0xcbadf720: Class-map: skinny_ssl, Inspect: skinny
            Active sess 1, most sess 1, byte 42916

hostname(config-tlsp)# show tls-proxy session count
2 in use, 4 most used

hostname(config-tlsp)# show tls-proxy session
2 in use, 4 most used
outside 133.9.0.211:50437 inside 195.168.2.200:2443 P:0xcbadf720(proxy) S:0xcbc48a08 byte
42940
outside 133.9.0.218:49159 inside 195.168.2.201:5061 P:0xcbae1538(sip_proxy) S:0xcbad5120
byte 8786

hostname(config-tlsp)# show tls-proxy session detail
2 in use, 4 most used
outside 133.9.0.211:50437 inside 195.168.2.200:2443 P:0xcbadf720(proxy) S:0xcbc48a08 byte
42940
    Client: State SSLOK  Cipher AES128-SHA Ch 0xca55e498 TxQSize 0 LastTxLeft 0 Flags 0x1
    Server: State SSLOK  Cipher AES128-SHA Ch 0xca55e478 TxQSize 0 LastTxLeft 0 Flags 0x9
Local Dynamic Certificate
    Status: Available
    Certificate Serial Number: 29
    Certificate Usage: General Purpose
    Public Key Type: RSA (1024 bits)
    Issuer Name:
        cn=TLS-Proxy-Signer
    Subject Name:
        cn=SEP0002B9EB0AAD
        o=Cisco Systems Inc
        c=US
    Validity Date:
        start date: 09:25:41 PDT Apr 16 2007
        end   date: 09:25:41 PDT Apr 15 2008
    Associated Trustpoints:

outside 133.9.0.218:49159 inside 195.168.2.201:5061 P:0xcbae1538(sip_proxy) S:0xcbad5120
byte 8786
    Client: State SSLOK  Cipher AES128-SHA Ch 0xca55e398 TxQSize 0 LastTxLeft 0 Flags 0x1
```

```
    Server: State SSLOK  Cipher AES128-SHA Ch 0xca55e378 TxQSize 0 LastTxLeft 0 Flags 0x9
Local Dynamic Certificate
    Status: Available
    Certificate Serial Number: 2b
    Certificate Usage: General Purpose
    Public Key Type: RSA (1024 bits)
    Issuer Name:
        cn=F1-ASA.default.domain.invalid
    Subject Name:
        cn=SEP0017593F50A8
    Validity Date:
        start date: 23:13:47 PDT Apr 16 2007
        end   date: 23:13:47 PDT Apr 15 2008
    Associated Trustpoints:
```

CTL Client

The CTL Client application supplied by Cisco Unified CallManager Release 5.1 and later supports a TLS proxy server (firewall) in the CTL file. Figure 27-3 through Figure 27-6 illustrate the TLS proxy features supported in the CTL Client.

Figure 27-3 CTL Client TLS Proxy Features — Add Firewall

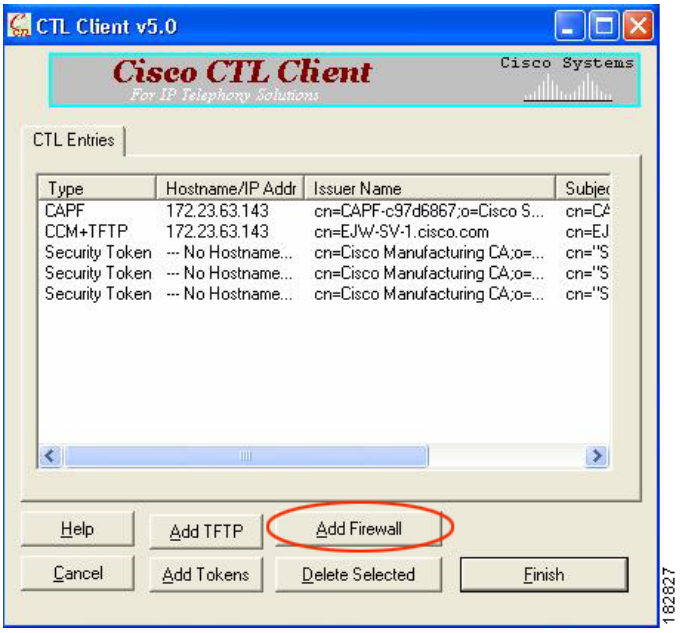


Figure 27-3 shows support for adding a CTL entry consisting of the security appliance as the TLS proxy.

Figure 27-4 CTL Client TLS Proxy Features — ASA IP Address or Domain Name

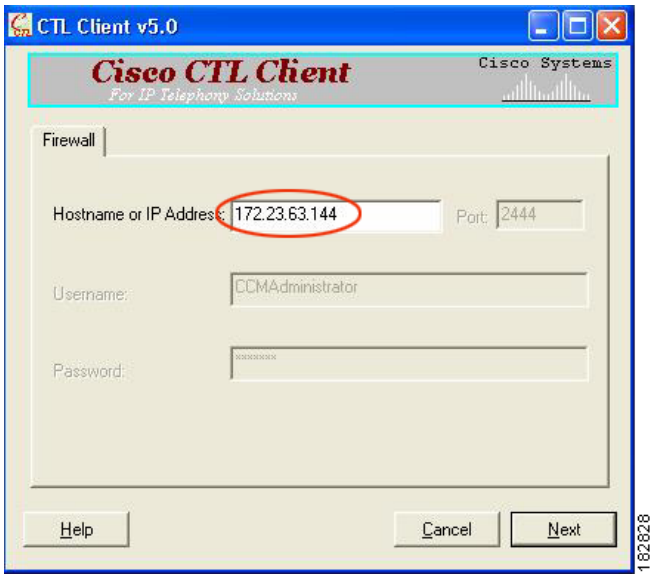


Figure 27-4 shows support for entering the security appliance IP address or domain name in the CTL Client.

Figure 27-5 CTL Client TLS Proxy Features — CTL Entry for ASA

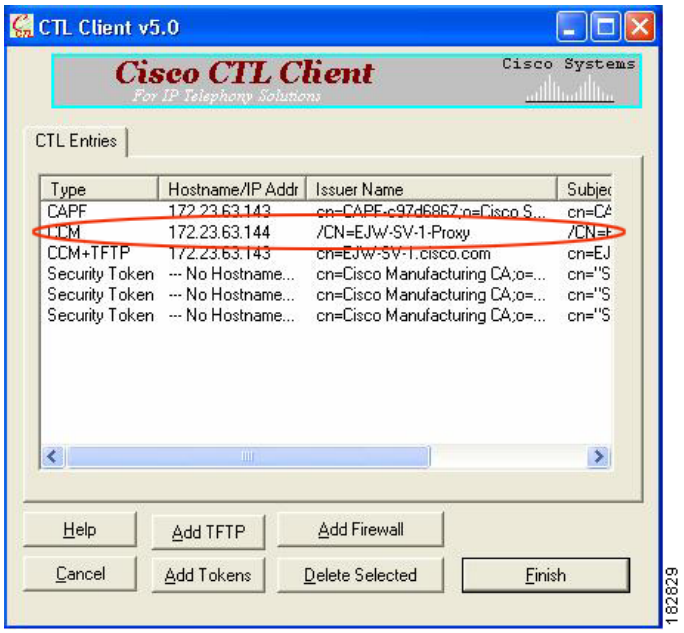
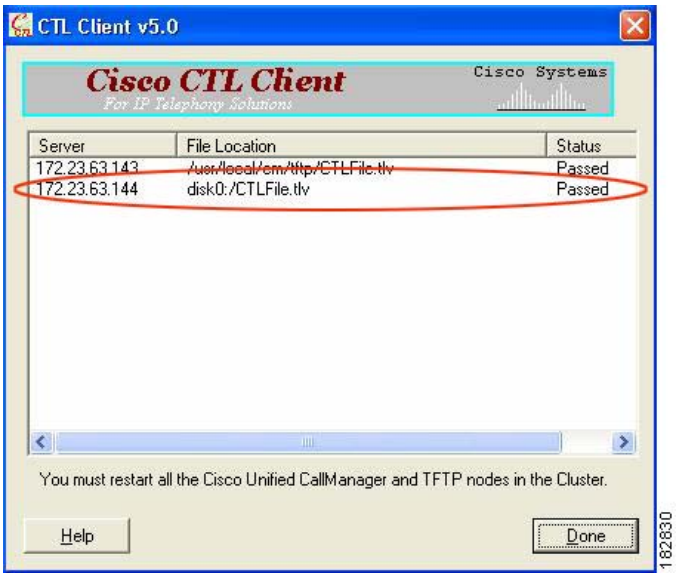


Figure 27-5 shows that the CTL entry for the security appliance as the TLS proxy has been added. The CTL entry is added after the CTL Client connects to the CTL Provider service on the security appliance and retrieves the proxy certificate.

Figure 27-6 CTL Client TLS Proxy Features — CTL File Installed on the ASA



The security appliance does not store the raw CTL file in the flash, rather, it parses the CTL file and installs appropriate trustpoints. Figure 27-6 indicates the installation was successful.

Cisco Unified Mobility and MMP Inspection Engine

This section includes the following topics:

- Mobility Proxy Overview, page 27-52
- Configuring the Security Appliance for Cisco Unified Mobility, page 27-57
- Debugging for Cisco Unified Mobility, page 27-58

Mobility Proxy Overview

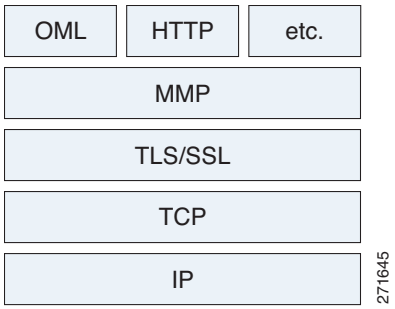
To support Cisco UMA for the Cisco Unified Mobility solution, the mobility proxy (implemented as a TLS proxy) includes the following functionality:

- The ability to allow no client authentication during the handshake with clients.
- Allowing an imported PKCS-12 certificate to server as a proxy certificate.

The security appliance includes an inspection engine to validate the Cisco UMA Mobile Multiplexing Protocol (MMP).

MMP is a data transport protocol for transmitting data entities between Cisco UMA clients and servers. As shown in Figure 27-7, MMP must be run on top of a connection-oriented protocol (the underlying transport) and is intended to be run on top of a secure transport protocol such as TLS. The Orative Markup Language (OML) protocol is intended to be run on top of MMP for the purposes of data synchronization, as well as the HTTP protocol for uploading and downloading large files.

Figure 27-7 MMP Stack



The TCP/TLS default port is 5443. There are no embedded NAT or secondary connections.

Cisco UMA client and server communications can be proxied via TLS, which decrypts the data, passes it to the inspect MMP module, and re-encrypt the data before forwarding it to the endpoint. The inspect MMP module verifies the integrity of the MMP headers and passes the OML/HTTP to an appropriate handler. The security appliance takes the following actions on the MMP headers and data:

- Verifies that client MMP headers are well-formed. Upon detection of a malformed header, the TCP session is terminated.
- Verifies that client to server MMP header lengths are not exceeded. If an MMP header length is exceeded (4096), then the TCP session is terminated.
- Verifies that client to server MMP content lengths are not exceeded. If an entity content length is exceeded (4096), the TCP session is terminated.

Note 4096 is the value currently used in MMP implementations.

Because MMP headers and entities can be split across packets, the security appliance buffers data to ensure consistent inspection. The SAPI (stream API) handles data buffering for pending inspection opportunities. MMP header text is treated as case insensitive and a space is present between header text and values. Reclaiming of MMP state is performed by monitoring the state of the TCP connection.

Mobility Proxy Deployment Scenarios

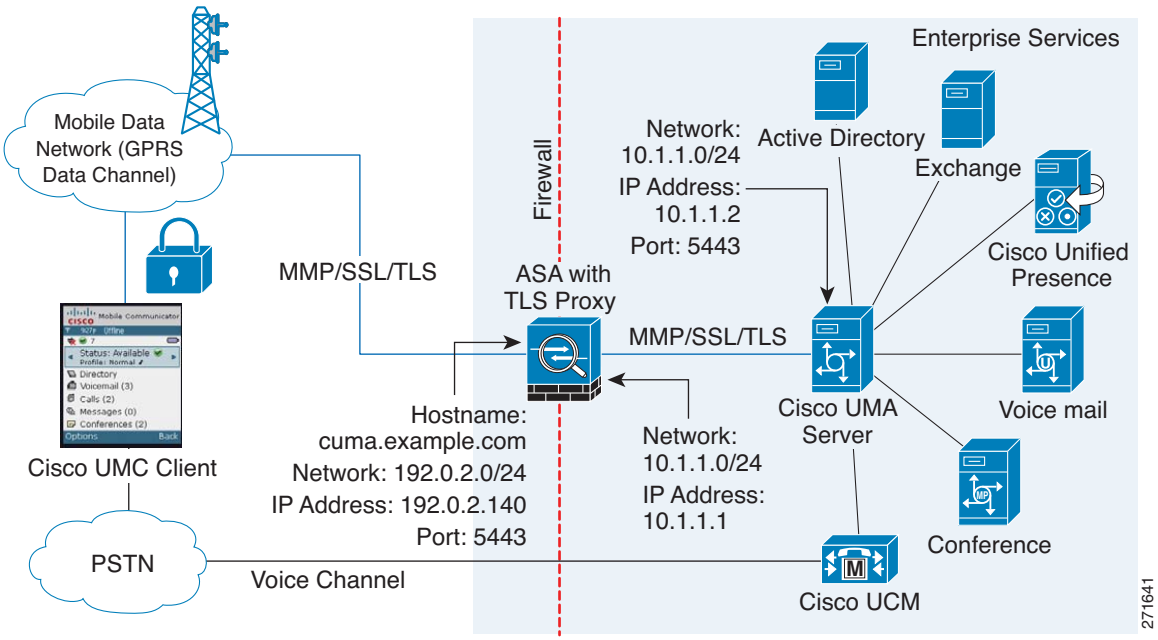
Figure 27-8 and Figure 27-9 show the two deployment scenarios for the TLS proxy used by the Cisco Unified Mobility solution. In scenario 1 (the recommended deployment architecture), the security appliance functions as both the firewall and TLS proxy. In scenario 2, the security appliance functions as the TLS proxy only and works with an existing firewall. In both scenarios, the clients connect from the Internet.

In the scenario 1 deployment, the security appliance is between a Cisco UMA client and a Cisco UMA server. The Cisco UMA client is an executable that is downloaded to each smartphone. The Cisco UMA client applications establishes a data connection, which is a TLS connection, to the corporate Cisco UMA server. The security appliance intercepts the connections and inspects the data that the client sends to the Cisco UMA server.

Note The TLS proxy for the Cisco Unified Mobility solution does not support client authentication because the Cisco UMA client cannot present a certificate. The following commands can be used to disable authentication during the TLS handshake.

```
hostname(config)# tls-proxy my_proxy
hostname(config-tlsp)# no server authenticate-client
```

Figure 27-8 Security Appliance as Firewall with Mobility Proxy and MMP Inspection



In Figure 27-8, the security appliance performs static NAT by translating the Cisco UMA server 10.1.1.2 IP address to 192.0.2.140.

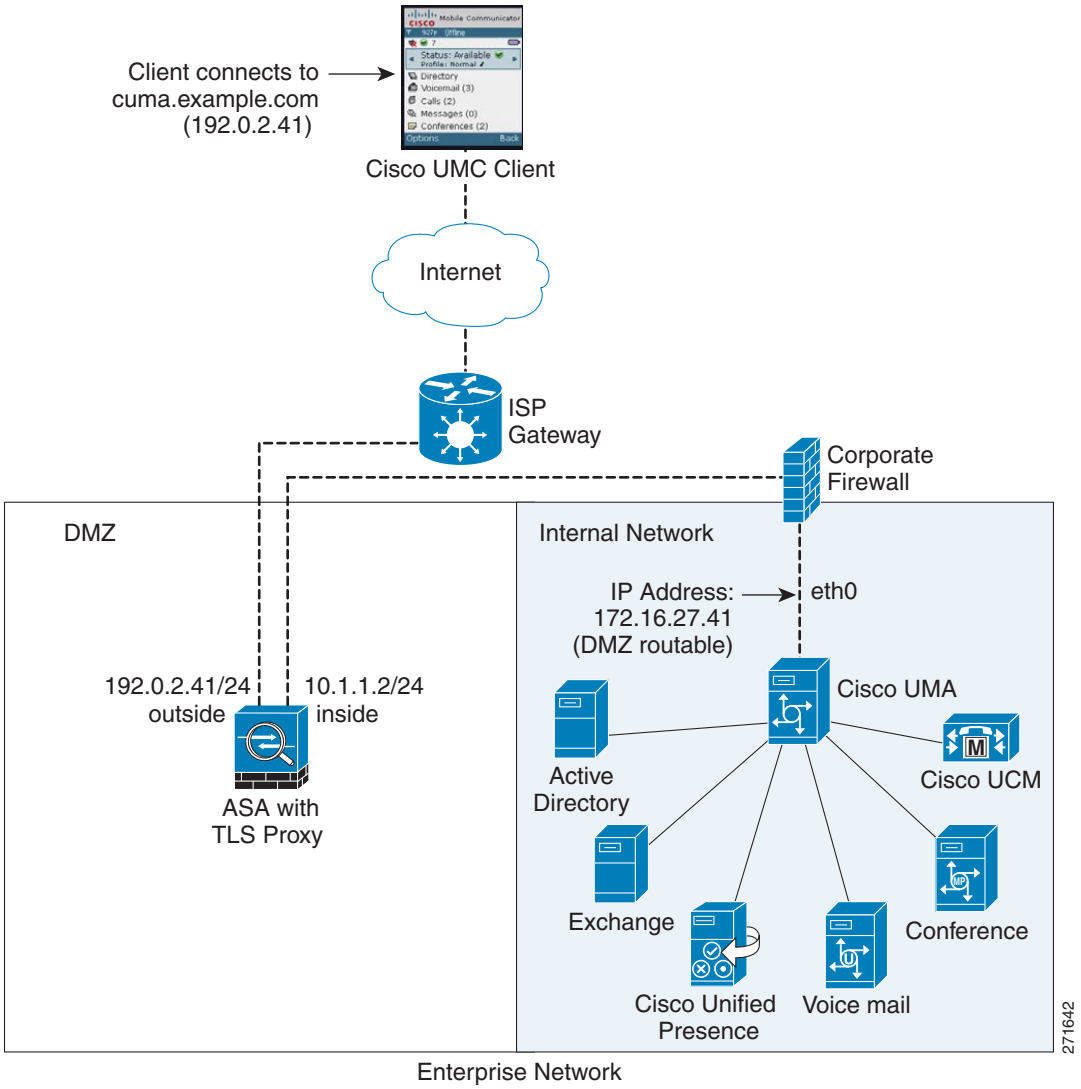
Figure 27-9 shows deployment scenario 2, where the security appliance functions as the TLS proxy only and does not function as the corporate firewall. In this scenario, the security appliance and the corporate firewall are performing NAT. The corporate firewall will not be able to predict which client from the Internet needs to connect to the corporate Cisco UMA server. Therefore, to support this deployment, you can take the following actions:

- Set up a NAT rule for inbound traffic that translates the destination IP address 192.0.2.41 to 172.16.27.41.
- Set up an interface PAT rule for inbound traffic translating the source IP address of every packet so that the corporate firewall does not need to open up a wildcard pinhole. The Cisco UMA server receives packets with the source IP address 192.0.12.183.

```
hostname(config)# nat (outside) 1 0.0.0.0 0.0.0.0 outside
hostname(config)# global (inside) 1 192.0.2.183 netmask 255.255.255.255
```

Note This interface PAT rule converges the Cisco UMA client IP addresses on the outside interface of the security appliance into a single IP address on the inside interface by using different source ports. Performing this action is often referred as “outside PAT”. “Outside PAT” is not recommended when TLS proxy for Cisco Unified Mobility is enabled on the same interface of the security appliance with phone proxy, Cisco Unified Presence, or any other features involving application inspection. “Outside PAT” is not supported completely by application inspection when embedded address translation is needed.

Figure 27-9 Cisco UMC/Cisco UMA Architecture – Scenario 2: Security Appliance as Mobility Proxy Only



Mobility Proxy Using NAT/PAT

In both scenarios (Figure 27-8 and Figure 27-9), NAT can be used to hide the private address of the Cisco UMA servers.

In scenario 2 (Figure 27-9), PAT can be used to converge all client traffic into one source IP, so that the firewall does not have to open up a wildcard pinhole for inbound traffic.

```
hostname(config)# access-list cumc extended permit tcp any host 172.16.27.41 eq 5443
```

versus

```
hostname(config)# access-list cumc extended permit tcp host 192.0.2.183 host 172.16.27.41
eq 5443
```

Establishing Trust Relationships for Cisco UMA Deployments

To establish a trust relationship between the Cisco UMC client and the security appliance, the security appliance uses the Cisco UMA server certificate and keypair or the security appliance obtains a certificate with the Cisco UMA server FQDN (certificate impersonation). Between the security appliance and the Cisco UMA server, the security appliance and Cisco UMA server use self-signed certificates or certificates issued by a local certificate authority.

Figure 27-10 shows how you can import the Cisco UMA server certificate onto the security appliance. When the Cisco UMA server has already enrolled with a third-party CA, you can import the certificate with the private key onto the security appliance. Then, the security appliance has the full credentials of the Cisco UMA server. When a Cisco UMA client connects to the Cisco UMA server, the security appliance intercepts the handshake and uses the Cisco UMA server certificate to perform the handshake with the client. The security appliance also performs a handshake with the server.

Figure 27-10 How the Security Appliance Represents Cisco UMA – Private Key Sharing

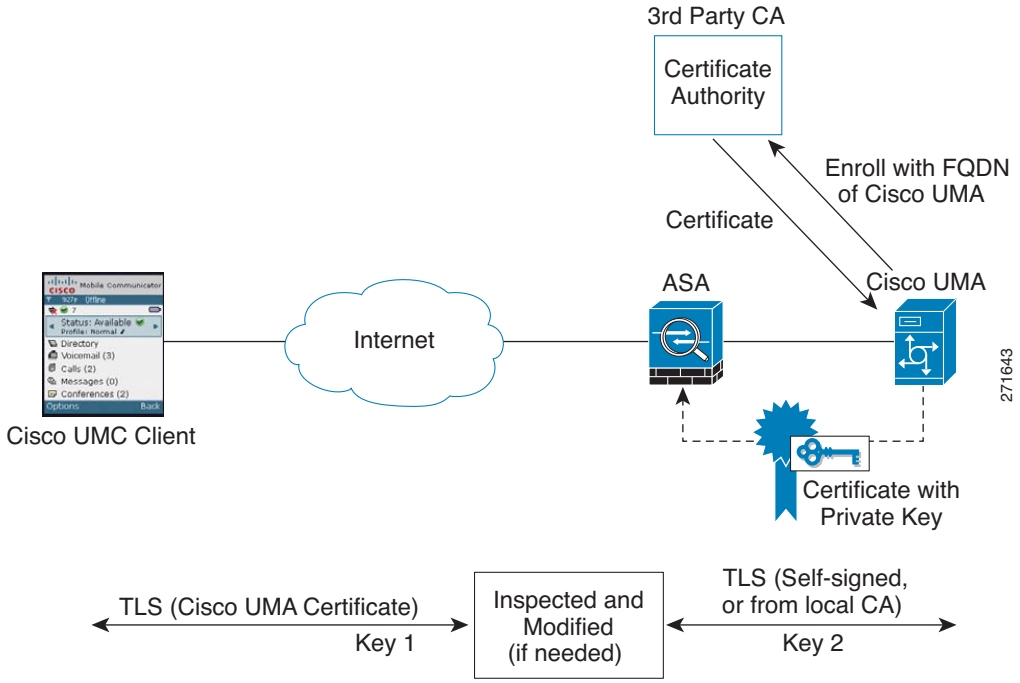
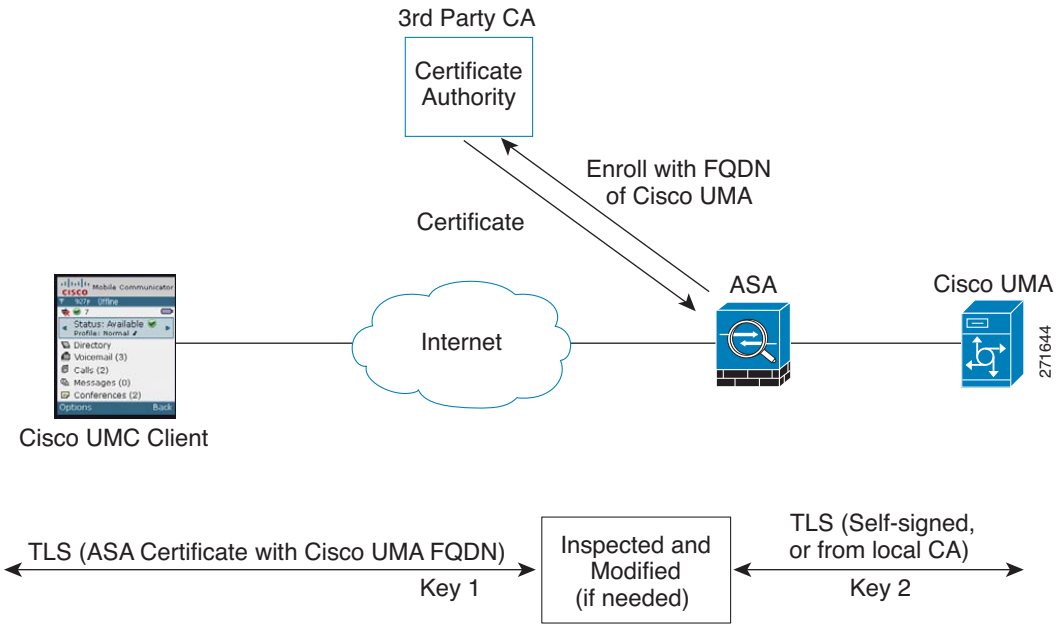


Figure 27-11 shows another way to establish the trust relationship. Figure 27-11 shows a green field deployment, because each component of the deployment has been newly installed. The security appliance enrolls with the third-party CA by using the Cisco UMA server FQDN as if the security appliance is the Cisco UMA server. When the Cisco UMA client connects to the security appliance, the security appliance presents the certificate that has the Cisco UMA server FQDN. The Cisco UMA client believes it is communicating to with the Cisco UMA server.

Figure 27-11 How the Security Appliance Represents Cisco UMA – Certificate Impersonation



A trusted relationship between the security appliance and the Cisco UMA server can be established with self-signed certificates. The security appliance's identity certificate is exported, and then uploaded on the Cisco UMA server truststore. The Cisco UMA server certificate is downloaded, and then uploaded on the security appliance truststore by creating a trustpoint and using the **crypto ca authenticate** command.

Configuring the Security Appliance for Cisco Unified Mobility

To configure for the security appliance to perform TLS proxy and MMP inspection as shown in Figure 27-8 and Figure 27-9, perform the following steps. It is assumed that self-signed certificates are used between the security appliance and the Cisco UMA server.

Step 1 Create the static NAT for the Cisco UMA server by entering the following command:

```
hostname(config)# static (real_ifc,mapped_ifc) mapped_ip real_ip netmask mask
```

Step 2 Export the Cisco UMA server certificate and keypair in PKCS-12 format. Import it onto the security appliance. The certificate will be used during the handshake with the Cisco UMA clients.

```
hostname(config)# crypto ca import trustpoint pkcs12 passphrase
[paste base 64 encoded pkcs12]
hostname(config)# quit
```

Step 3 Install the Cisco UMA server self-signed certificate in the security appliance truststore. This step is necessary for the security appliance to authenticate the Cisco UMA server during the handshake between the security appliance proxy and Cisco UMA server.

```
hostname(config)# crypto ca trustpoint trustpoint_name
hostname(config-ca-trustpoint)# enrollment terminal
hostname(config-ca-trustpoint)# exit
hostname(config)# crypto ca authenticate trustpoint
hostname(config)# Enter the base 64 encoded CA certificate.
hostname(config)# End with a blank line or the word "quit" on a line by itself
```

```
  [ certificate data omitted ]
hostname(config)# quit
```

Step 4 Create a TLS proxy instance for the Cisco UMA clients connecting to the Cisco UMA server:

```
hostname(config)# tls-proxy proxy_name
hostname(config-tlsp)# server trust-point proxy_name
hostname(config-tlsp)# client trust-point proxy_name
hostname(config-tlsp)# no server authenticate-client
hostname(config-tlsp)# client cipher-suite cipher_suite
```

Step 5 Enable the TLS proxy for MMP inspection:

```
hostname(config)# class-map class_map_name
hostname(config-cmap)# match port tcp eq port
hostname(config-cmap)# exit
hostname(config)# policy-map name
hostname(config-pmap)# class name
hostname(config-pmap)# inspect mmp tls-proxy proxy_name
hostname(config-pmap)# exit
hostname(config)# service-policy policy_map_name global
```

Debugging for Cisco Unified Mobility

Mobility proxy can be debugged the same way as IP Telephony. You can enable TLS proxy debug flags along with SSL syslogs to debug TLS proxy connection problems.

For example, using the following commands to enable TLS proxy-related debugging and syslog output only:

```
hostname# debug inspect tls-proxy events
hostname# debug inspect tls-proxy errors
hostname# config terminal
hostname(config)# logging enable
hostname(config)# logging timestamp
hostname(config)# logging list loglist message 711001
hostname(config)# logging list loglist message 725001-725014
hostname(config)# logging list loglist message 717001-717038
hostname(config)# logging buffer-size 1000000
hostname(config)# logging buffered loglist
hostname(config)# logging debug-trace
```

For information about TLS proxy debugging techniques and sample output, see TLS Proxy for Encrypted Voice Inspection, page 27-42.

Enable the **debug mmp** command for MMP inspection engine debugging:

```
MMP:: received 60 bytes from outside:1.1.1.1/2000 to inside:2.2.2.2/5443
MMP:: version OLWP-2.0
MMP:: forward 60/60 bytes from outside:1.1.1.1/2000 to inside:2.2.2.2/5443
MMP:: received 100 bytes from inside:2.2.2.2/5443 to outside:1.1.1.1/2000
MMP:: session-id: ABCD_1234
MMP:: status: 201
MMP:: forward 100/100 bytes from inside:2.2.2.2/5443 to outside 1.1.1.1/2000
MMP:: received 80 bytes from outside:1.1.1.1/2000 to inside:2.2.2.2/5443
MMP:: content-type: http/1.1
MMP:: content-length: 40
```

You can also capture the raw and decrypted data by the TLS proxy by entering the following commands:

```
hostname# capture mycap interface outside (capturing raw packets)
```

```
hostname# capture mycap-dec type tls-proxy interface outside (capturing decrypted data)
hostname# show capture capture_name
hostname# copy /pcap capture:capture_name tftp://tftp_location
```

Cisco Unified Presence

This section includes the following topics:

- Architecture for Cisco Unified Presence, page 27-59
- Configuring the Presence Federation Proxy for Cisco Unified Presence, page 27-62
- Debugging the Security Appliance for Cisco Unified Presence, page 27-64

Architecture for Cisco Unified Presence

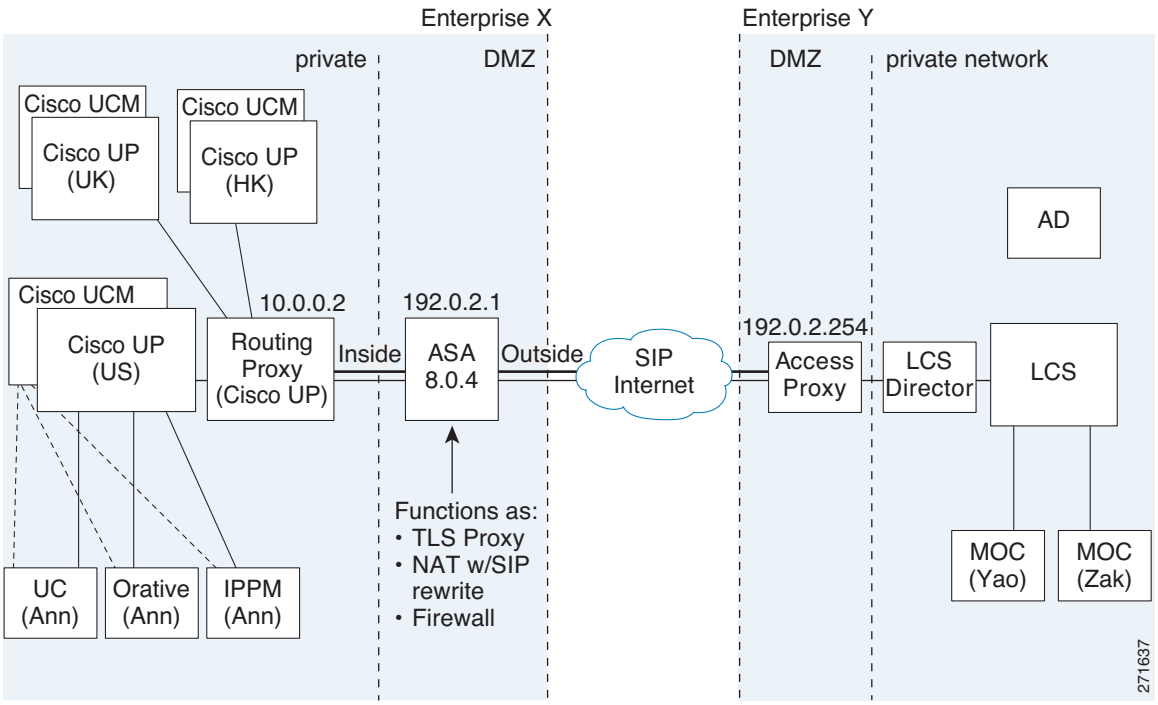
Figure 27-12 depicts a Cisco Unified Presence/LCS Federation scenario with the security appliance as the presence federation proxy (implemented as a TLS proxy). The two entities with a TLS connection are the "Routing Proxy" (a dedicated Cisco UP) in Enterprise X and the Microsoft Access Proxy in Enterprise Y. However, the deployment is not limited to this scenario. Any Cisco UP or Cisco UP cluster could be deployed on the left side of the security appliance; the remote entity could be any server (an LCS, an OCS, or another Cisco UP).

The following architecture is generic for two servers using SIP (or other security appliance inspected protocols) with a TLS connection.

Entity X: Cisco UP/Routing Proxy in Enterprise X

Entity Y: Microsoft Access Proxy/Edge server for LCS/OCS in Enterprise Y

Figure 27-12 Typical Cisco Unified Presence/LCS Federation Scenario



In the above architecture, the security appliance functions as a firewall, NAT, and TLS proxy, which is the recommended architecture. However, the security appliance can also function as NAT and the TLS proxy alone, working with an existing firewall.

Either server can initiate the TLS handshake (unlike IP Telephony or Cisco Unified Mobility, where only the clients initiate the TLS handshake). There are by-directional TLS proxy rules and configuration. Each enterprise can have an security appliance as the TLS proxy.

In Figure 27-12, NAT or PAT can be used to hide the private address of Entity X. In this situation, static NAT or PAT must be configured for foreign server (Entity Y) initiated connections or the TLS handshake (inbound). Typically, the public port should be 5061. The following static PAT command is required for the Cisco UP that accepts inbound connections:

```
hostname(config)# static (inside,outside) tcp 192.0.2.1 5061 10.0.0.2 5061 netmask
255.255.255.255
```

The following static PAT must be configured for each Cisco UP that could initiate a connection (by sending SIP SUBSCRIBE) to the foreign server.

For Cisco UP with the address 10.0.0.2, enter the following command:

```
hostname(config)# static (inside,outside) tcp 192.0.2.1 5062 10.0.0.2 5062 netmask
255.255.255.255
hostname(config)# static (inside,outside) udp 192.0.2.1 5070 10.0.0.2 5070 netmask
255.255.255.255
hostname(config)# static (inside,outside) tcp 192.0.2.1 5060 10.0.0.2 5060 netmask
255.255.255.255
```

For another Cisco UP with the address 10.0.0.3, you must use a different set of PAT ports, such as 45062 or 45070:

```
hostname(config)# static (inside,outside) tcp 192.0.2.1 45061 10.0.0.3 5061 netmask
255.255.255.255
```

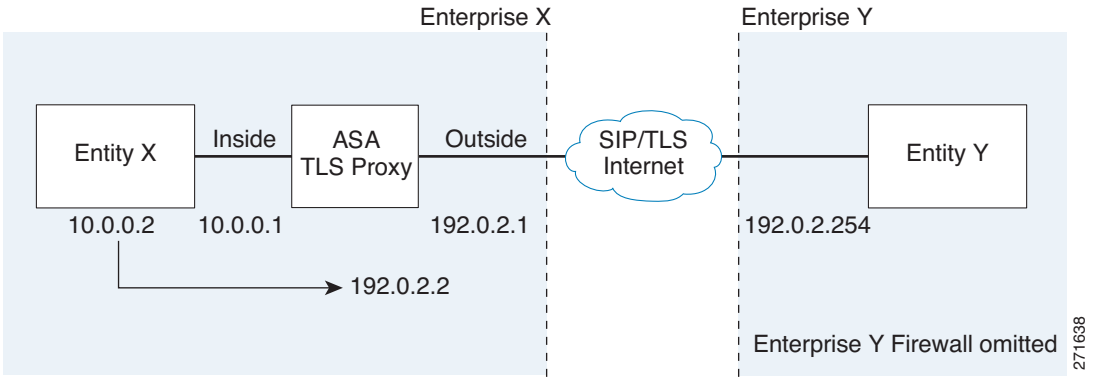
```
hostname(config)# static (inside,outside) tcp 192.0.2.1 45062 10.0.0.3 5062 netmask
255.255.255.255
hostname(config)# static (inside,outside) udp 192.0.2.1 45070 10.0.0.3 5070 netmask
255.255.255.255
hostname(config)# static (inside,outside) tcp 192.0.2.1 5070 10.0.0.2 5070 netmask
255.255.255.255
hostname(config)# static (inside,outside) tcp 192.0.2.1 45060 10.0.0.3 5060 netmask
255.255.255.255
```

Dynamic NAT or PAT can be used for the rest of the outbound connections or the TLS handshake. The security appliance SIP inspection engine takes care of the necessary translation (fixup).

```
hostname(config)# global (outside) 102 192.0.2.1 netmask 255.255.255.255
hostname(config)# nat (inside) 102 0.0.0.0 0.0.0.0
```

Figure 27-13 illustrates an abstracted scenario with Entity X connected to Entity Y through the presence federation proxy on the security appliance. The proxy is in the same administrative domain as Entity X. Entity Y could have another security appliance as the proxy but this is omitted for simplicity.

Figure 27-13 Abstracted Presence Federation Proxy Scenario between Two Server Entities



For the Entity X domain name to be resolved correctly when the security appliance holds its credential, the security appliance could be configured to perform NAT for Entity X, and the domain name is resolved as the Entity X public address for which the security appliance provides proxy service.

Establishing a Trust Relationship in the Presence Federation

Within an enterprise, setting up a trust relationship is achievable by using self-signed certificates or you can set it up on an internal CA.

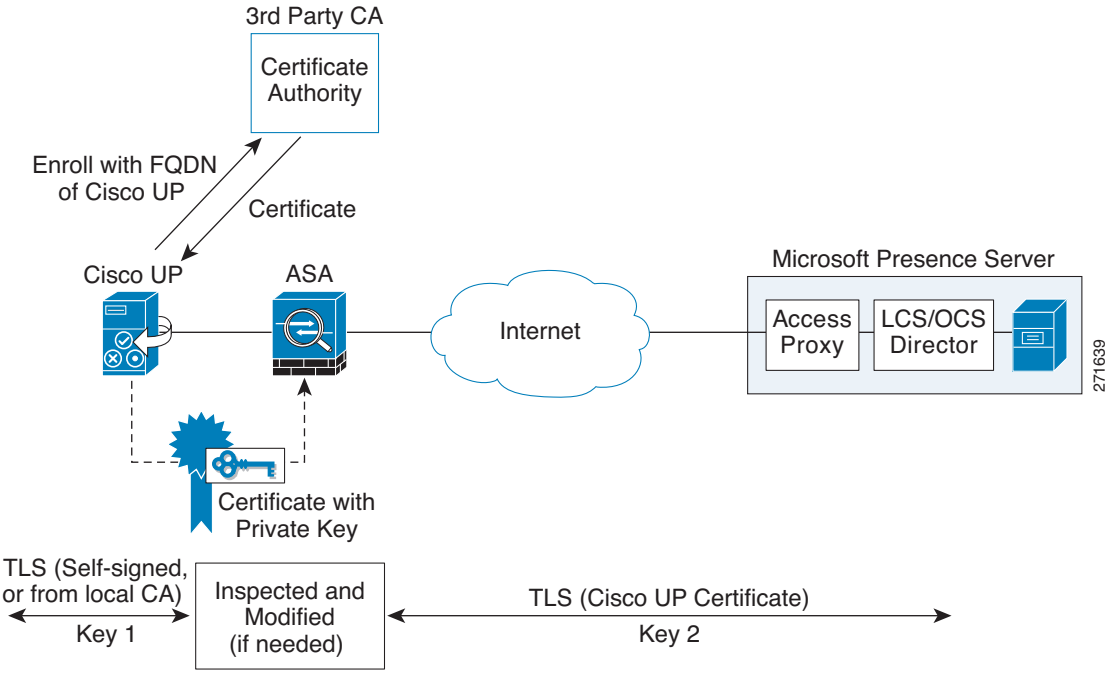
Establishing a trust relationship cross enterprises or across administrative domains is key for federation. Cross enterprises you must use a trusted third-party CA (such as, VeriSign). The security appliance obtains a certificate with the FQDN of the Cisco UP (certificate impersonation).

For the TLS handshake, the two entities could validate the peer certificate via a certificate chain to trusted third-party certificate authorities. Both entities enroll with the CAs. The security appliance as the TLS proxy must be trusted by both entities. The security appliance is always associated with one of the enterprises. Within that enterprise (Enterprise X in Figure 27-12), the entity and the security appliance could authenticate each other via a local CA, or by using self-signed certificates.

To establish a trusted relationship between the security appliance and the remote entity (Entity Y), the security appliance can enroll with the CA on behalf of Entity X (Cisco UP). In the enrollment request, the Entity X identity (domain name) is used.

Figure 27-14 shows the way to establish the trust relationship. The security appliance enrolls with the third party CA by using the Cisco UP FQDN as if the security appliance is the Cisco UP.

Figure 27-14 How the Security Appliance Represents Cisco Unified Presence – Certificate Impersonate



About the Security Certificate Exchange Between Cisco UP and the Security Appliance

You need to generate the keypair for the certificate (such as `cup_proxy_key`) used by the security appliance, and configure a trustpoint to identify the self-signed certificate sent by the security appliance to Cisco UP (such as `cup_proxy`) in the TLS handshake.

For the security appliance to trust the Cisco UP certificate, you need to create a trustpoint to identify the certificate from the Cisco UP (such as `cert_from_cup`), and specify the enrollment type as terminal to indicate that you will paste the certificate received from the Cisco UP into the terminal.

Configuring the Presence Federation Proxy for Cisco Unified Presence

To configure a Cisco Unified Presence/LCS Federation scenario with the security appliance as the TLS proxy where there is a single Cisco UP that is in the local domain and self-signed certificates are used between the Cisco UP and the security appliance (like the scenario shown in Figure 27-12), perform the following steps.

Step 1 Create the following static NAT for the local domain containing the Cisco UP.

For the inbound connection to the local domain containing the Cisco UP, create static PAT by entering the following command:

```
hostname(config)# static (real_ifc,mapped_ifc) tcp mapped_ip mapped_port netmask mask
```

Note For each Cisco UP that could initiate a connection (by sending SIP SUBSCRIBE) to the foreign server, you must also configure static PAT by using a different set of PAT ports.

For outbound connections or the TLS handshake, use dynamic NAT or PAT. The security appliance SIP inspection engine takes care of the necessary translation (fixup).

```
hostname(config)# global (mapped_ifc) nat_id mapped_ip netmask mask
hostname(config)# nat (real_ifc) nat_id real_ip mask
```

Step 2 Create the necessary RSA keypairs by entering the following command:

```
hostname(config)# crypto key generate rsa label key-pair-label modulus size
```

The keypair is used by the self-signed certificate presented to the local domain containing the Cisco UP (proxy for the remote entity).

Step 3 Create a proxy certificate, which is a self-signed certificate, for the remote entity by entering the following commands:

```
hostname(config)# crypto ca trustpoint trustpoint_name
hostname(config-ca-trustpoint)# enrollment self
hostname(config-ca-trustpoint)# fqdn none
hostname(config-ca-trustpoint)# subject-name X.500_name
hostname(config-ca-trustpoint)# keypair keyname
hostname(config-ca-trustpoint)# exit
hostname(config)# crypto ca enroll trustpoint
```

You will install the certificate on the local entity truststore. You could also enroll the certificate with a local CA trusted by the local entity.

Step 4 Export the self-signed certificate for the security appliance created in Step 3 and install it as a trusted certificate on the local entity. This step is necessary for local entity to authenticate the security appliance.

Export the security appliance self-signed (identity) certificate by entering the following command:

```
hostname(config)# crypto ca export trustpoint identity-certificate
```

Step 5 Export the local entity certificate and install it on the security appliance by entering the following commands. This step is needed for the security appliance to authenticate the local entity during the handshake. If the local entity uses a self-signed certificate, the self-signed certificate must be installed; if the local entity uses a CA-issued certificate, the CA certificate needs to be installed. The following configuration shows the commands for using a self-signed certificate.

```
hostname(config)# crypto ca trustpoint trustpoint_name
hostname(config-ca-trustpoint)# enrollment terminal
hostname(config-ca-trustpoint)# exit
hostname(config)# crypto ca authenticate trustpoint
hostname(config)# Enter the base 64 encoded CA certificate.
hostname(config)# End with a blank line or the word "quit" on a line by itself
  [ certificate data omitted ]
hostname(config)# quit
```

Step 6 To create a proxy certificate on the security appliance that is trusted by the remote entity, obtain a certificate from a trusted CA. For information about obtaining a certificate from a trusted CA, see Certificate Configuration, page 41-5.

Step 7 Install the CA certificate that signs the remote entity certificate on the security appliance by entering the following commands. This step is necessary for the security appliance to authenticate the remote entity.

```
hostname(config)# crypto ca trustpoint trustpoint_name
hostname(config-ca-trustpoint)# enrollment terminal
hostname(config-ca-trustpoint)# exit
```

```
hostname(config)# crypto ca authenticate trustpoint
hostname(config)# Enter the base 64 encoded CA certificate.
hostname(config)# End with a blank line or the word "quit" on a line by itself
  [ certificate data omitted ]
hostname(config)# quit
```

Step 8 Create TLS proxy instances for the local and remote entity initiated connections respectively. The entity that initiates the TLS connection is in the role of "TLS client". Because the TLS proxy has a strict definition of "client" and "server" proxy, two TLS proxy instances must be defined if either of the entities could initiate the connection.

```
! Local entity to remote entity
hostname(config)# tls-proxy proxy_name
hostname(config-tlsp)# server trust-point proxy_name
hostname(config-tlsp)# client trust-point proxy_trustpoint
hostname(config-tlsp)# client cipher-suite cipher_suite
```

Where the *proxy_name* for the **server trust-point** command is the remote entity proxy name and the *proxy_trustpoint* for the **client trust-point** command is the local entity proxy.

```
! Remote entity to local entity
hostname(config)# tls-proxy proxy_name
hostname(config-tlsp)# server trust-point proxy_name
hostname(config-tlsp)# client trust-point proxy_trustpoint
hostname(config-tlsp)# client cipher-suite cipher_suite
```

Where the *proxy_name* for the **server trust-point** command is the local entity proxy name and the *proxy_trustpoint* for the **client trust-point** command is the remote entity proxy.

Step 9 Enable the TLS proxy for SIP inspection and define policies for both entities that could initiate the connection by entering the following commands:

```
hostname(config)# access-list id extended permit tcp host src_ip host dest_ip eq port
hostname(config)# class-map class_map_name
hostname(config-cmap)# match access-list access_list_name
hostname(config-cmap)# exit
hostname(config)# policy-map type inspect sip policy_map_name
hostname(config-pmap)# parameters
    ! SIP inspection parameters
hostname(config-pmap)# exit
hostname(config)# policy-map name
hostname(config-pmap)# class name
hostname(config-pmap)# inspect sip sip_map tls-proxy proxy_name
hostname(config-pmap)# exit
hostname(config)# service-policy policy_map_name global
```

Where *name* for the **policy-map** command is the name of the global policy map.

Debugging the Security Appliance for Cisco Unified Presence

Debugging is similar to debugging TLS proxy for IP Telephony. You can enable TLS proxy debug flags along with SSL syslogs to debug TLS proxy connection problems.

For example, use the following commands to enable TLS proxy-related debug and syslog output only:

```
hostname(config)# debug inspect tls-proxy events
hostname(config)# debug inspect tls-proxy errors
hostname(config)# logging enable
hostname(config)# logging timestamp
hostname(config)# logging list loglist message 711001
```

```
hostname(config)# logging list loglist message 725001-725014
hostname(config)# logging list loglist message 717001-717038
hostname(config)# logging buffer-size 1000000
hostname(config)# logging buffered loglist
hostname(config)# logging debug-trace
```

For information about TLS proxy debugging techniques and sample output, see TLS Proxy for Encrypted Voice Inspection, page 27-42.

Enable the **debug sip** command for SIP inspection engine debugging. See the *Cisco Security Appliance Command Reference*.

Additionally, you can capture the raw and decrypted data by the TLS proxy by entering the following commands:

```
hostname# capture mycap interface outside (capturing raw packets)
hostname# capture mycap-dec type tls-proxy interface outside (capturing decrypted data)
hostname# show capture capture_name
hostname# copy /pcap capture:capture_name tftp://tftp_location
```

Sample Configurations for Cisco Unified Communications Proxy Features

This section includes the following topics:

- Phone Proxy Sample Configurations, page 27-65
- Cisco Unified Mobility Sample Configurations, page 27-75
- Cisco Unified Presence Sample Configuration, page 27-78

Phone Proxy Sample Configurations

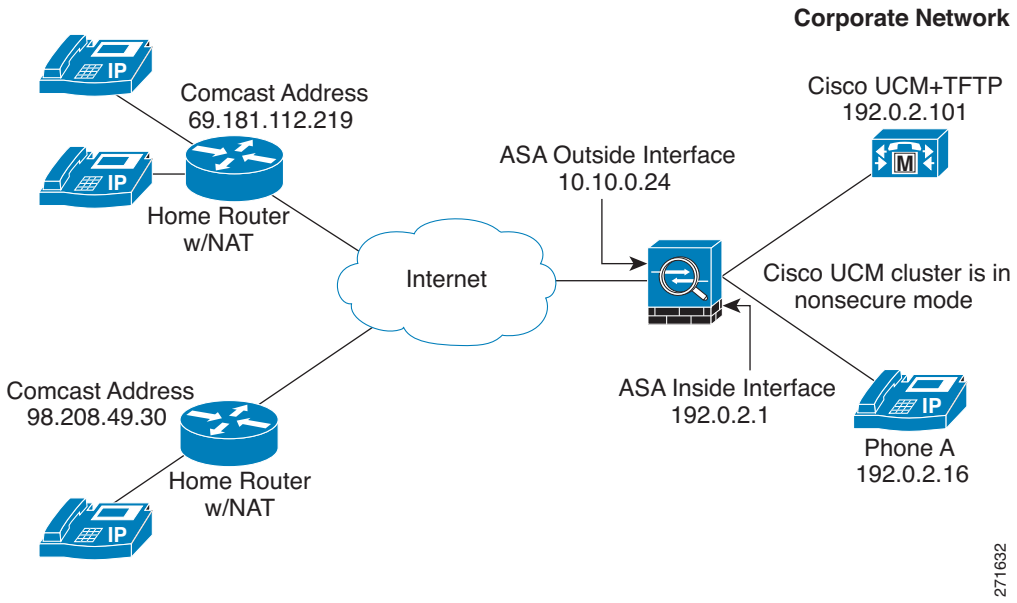
This section includes the following topics:

- Example 1: Nonsecure Cisco UCM cluster, Cisco UCM and TFTP Server on Publisher, page 27-65
- Example 2: Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Publisher, page 27-66
- Example 3: Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Different Servers, page 27-68
- Example 4: Mixed-mode Cisco UCM cluster, Primary Cisco UCM, Secondary and TFTP Server on Different Servers, page 27-69
- Example 5: LSC Provisioning in Mixed-mode Cisco UCM cluster; Cisco UCM and TFTP Server on Publisher, page 27-71
- Example 6: VLAN Transversal, page 27-73

Example 1: Nonsecure Cisco UCM cluster, Cisco UCM and TFTP Server on Publisher

Figure 27-15 shows an example of the configuration for a non-secure Cisco UCM cluster using the following topology.

Figure 27-15 Nonsecure Cisco UCM cluster, Cisco UCM and TFTP Server on Publisher

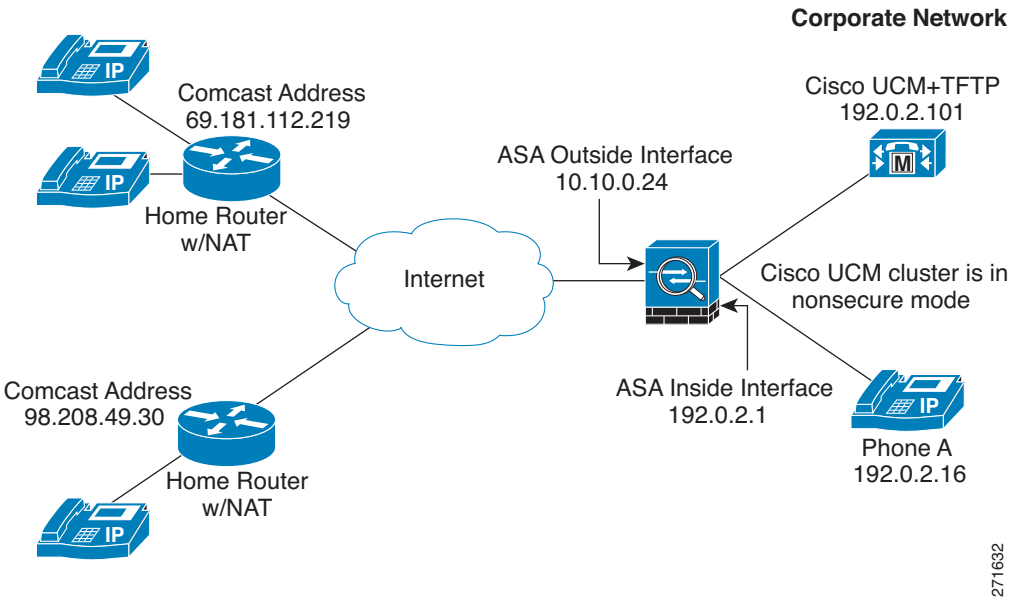



```
static (inside,outside) 10.10.0.26 192.0.2.101
access-list pp extended permit udp any host 10.10.0.26 eq 69
access-group pp in interface outside
crypto key generate rsa label cucmtftp_kp modulus 1024
crypto ca trustpoint cucm_tftp_server
    enrollment self
    keypair cucmtftp_kp
crypto ca enroll cucm_tftp_server
ctl-file myctl
    record-entry cucm-tftp trustpoint cucm_tftp_server address 10.10.0.26
    no shutdown
tls-proxy mytls
    server trust-point _internal_PP_myctl
phone-proxy mypp
    media-termination address 192.0.2.25
    tftp-server address 192.0.2.101 interface inside
    tls-proxy mytls
    ctl-file myctl
class-map sec_sccp
    match port tcp 2443
class-map sec_sip
    match port tcp eq 5061
policy-map pp_policy
    class sec_sccp
        inspect skinny phone-proxy mypp
    class sec_sip
        inspect sip phone-proxy mypp
service-policy pp_policy interface outside
```

Example 2: Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Publisher

Figure 27-16 shows an example of the configuration for a mixed-mode Cisco UCM cluster using the following topology.

Figure 27-16 Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Publisher




```
static (inside,outside) 10.10.0.26 192.0.2.101
access-list pp extended permit udp any host 10.10.0.26 eq 69
access-group pp in interface outside
crypto key generate rsa label cucmtftp_kp modulus 1024
crypto ca trustpoint cucm_tftp_server
    enrollment self
    keypair cucmtftp_kp
crypto ca enroll cucm_tftp_server
ctl-file myctl
    record-entry cucm-tftp trustpoint cucm_tftp_server address 10.10.0.26
    no shutdown
crypto key generate rsa label ldc_signer_key modulus 1024
crypto key generate rsa label phone_common modulus 1024
crypto ca trustpoint ldc_server
    enrollment self
    proxy_ldc_issuer
    fqdn my-ldc-ca.exmaple.com
    subject-name cn=FW_LDC_SIGNER_172_23_45_200
    keypair ldc_signer_key
    crypto ca enroll ldc_server
tls-proxy my_proxy
    server trust-point _internal_PP_myctl
    client ldc issuer ldc_server
    client ldc keypair phone_common
    client cipher-suite aes128-sha1 aes256-sha1
phone-proxy mypp
    media-termination address 10.10.0.25
    tftp-server address 192.0.2.101 interface inside
    tls-proxy mytls
    ctl-file myctl
    cluster-mode mixed
class-map sec_sccp
    match port tcp 2443
class-map sec_sip
    match port tcp eq 5061
policy-map pp_policy
    class sec_sccp
        inspect skinny phone-proxy mypp
```

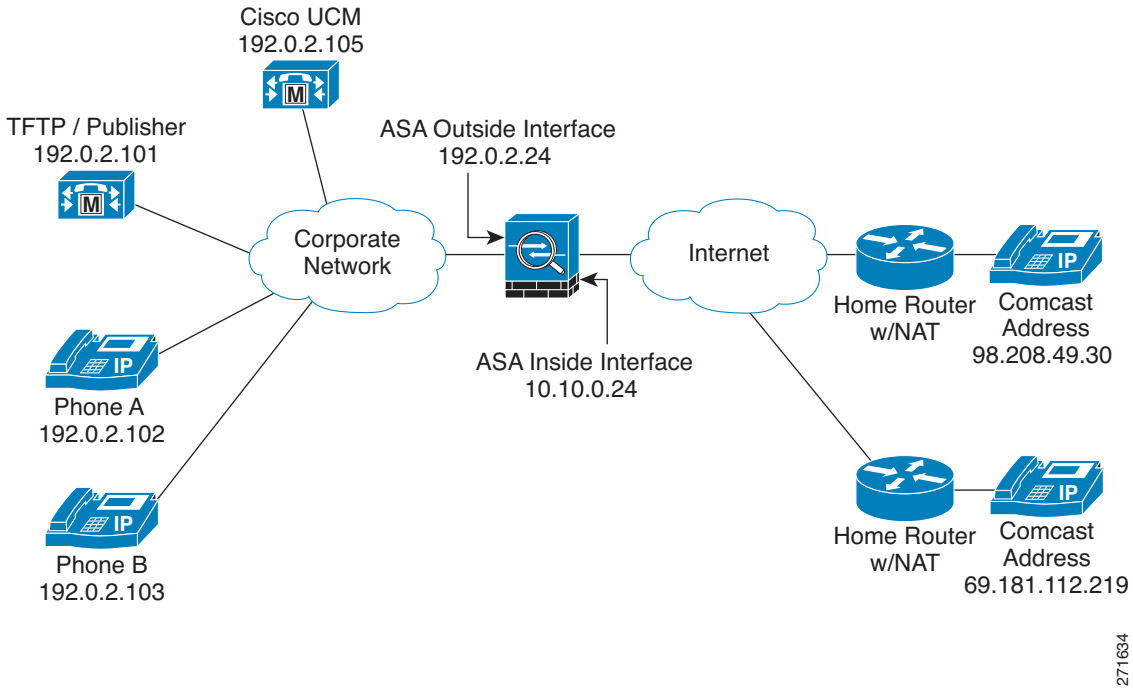
```
    class sec_sip
        inspect sip phone-proxy mypp
service-policy pp_policy interface outside
```

Example 3: Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Different Servers

Figure 27-17 shows an example of the configuration for a mixed-mode Cisco UCM cluster using the following topology where the TFTP server resides on a different server from the Cisco UCM.

In this sample, the static interface PAT for the TFTP server is configured to appear like the security appliance's outside interface IP address.

Figure 27-17 Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Different Servers




```
static (inside,outside) 10.10.0.26 192.0.2.105
static (inside,outside) udp interface 69 192.0.2.101 69
access-list pp extended permit udp any host 10.10.0.24 eq 69
access-group pp in interface outside
crypto key generate rsa label cucm_kp modulus 1024
crypto ca trustpoint cucm
    enrollment self
    keypair cucm_kp
crypto ca enroll cucm
crypto key generate rsa label tftp_kp modulus 1024
crypto ca trustpoint tftp_server
    enrollment self
    keypair tftp_kp
crypto ca enroll tftp_server
ctl-file myctl
    record-entry cucm trustpoint cucm_server address 10.10.0.26
    no shutdown
crypto key generate rsa label ldc_signer_key modulus 1024
crypto key generate rsa label phone_common modulus 1024
crypto ca trustpoint ldc_server
    enrollment self
```

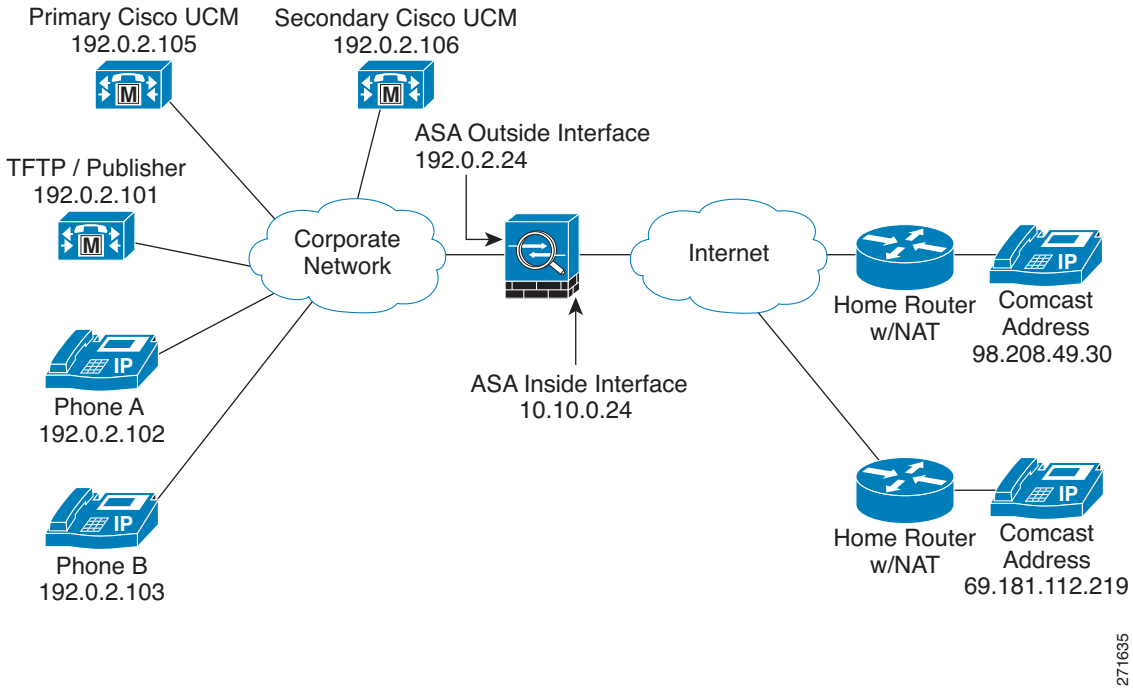
```
    proxy_ldc_issuer
    fqdn my-ldc-ca.exmaple.com
    subject-name cn=FW_LDC_SIGNER_172_23_45_200
    keypair ldc_signer_key
    crypto ca enroll ldc_server
tls-proxy my_proxy
    server trust-point _internal_PP_myctl
    client ldc issuer ldc_server
    client ldc keypair phone_common
    client cipher-suite aes128-sha1 aes256-sha1
phone-proxy mypp
    media-termination address 10.10.0.25
    tftp-server address 192.0.2.101 interface inside
    tls-proxy mytls
    ctl-file myctl
    cluster-mode mixed
class-map sec_sccp
    match port tcp 2443
class-map sec_sip
    match port tcp eq 5061
policy-map pp_policy
    class sec_sccp
        inspect skinny phone-proxy mypp
    class sec_sip
        inspect sip phone-proxy mypp
service-policy pp_policy interface outside
```

Example 4: Mixed-mode Cisco UCM cluster, Primary Cisco UCM, Secondary and TFTP Server on Different Servers

Figure 27-18 shows an example of the configuration for a mixed-mode Cisco UCM cluster using the following topology where the TFTP server resides on a different server from the primary and secondary Cisco UCMs.

In this sample, the static interface PAT for the TFTP server is configured to appear like the security appliance's outside interface IP address.

Figure 27-18 Mixed-mode Cisco UCM cluster, Primary Cisco UCM, Secondary Cisco UCM, and TFTP Server on Different Servers




```
static (inside,outside) 10.10.0.27 192.0.2.105
static (inside,outside) 10.10.0.26 192.0.2.106
static (inside,outside) udp interface 69 192.0.2.101 69
access-list pp extended permit udp any host 10.10.0.24 eq 69
access-group pp in interface outside
crypto key generate rsa label cluster_kp modulus 1024
crypto ca trustpoint pri_cucm
    enrollment self
    keypair cluster_kp
crypto ca enroll pri_cucm
crypto ca trustpoint sec_cucm
    enrollment self
    serial-number
    keypair cluster_kp
crypto ca enroll sec_cucm
crypto ca trustpoint tftp_server
    enrollment self
    fqdn my_tftp.example.com
    keypair cluster_kp
crypto ca enroll tftp_server
ctl-file myctl
    record-entry tftp trustpoint tftp_server address 10.10.0.24
    record-entry cucm trustpoint pri_cucm_server address 10.10.0.27
    record-entry cucm trustpoint sec_cucm_server address 10.10.0.2
    no shutdown
crypto key generate rsa label ldc_signer_key modulus 1024
crypto key generate rsa label phone_common modulus 1024
crypto ca trustpoint ldc_server
    enrollment self
    proxy_ldc_issuer
    fqdn my-ldc-ca.exmaple.com
    subject-name cn=FW_LDC_SIGNER_172_23_45_200
    keypair ldc_signer_key
```

```
    crypto ca enroll ldc_server
tls-proxy my_proxy
    server trust-point _internal_PP_myctl
    client ldc issuer ldc_server
    client ldc keypair phone_common
    client cipher-suite aes128-sha1 aes256-sha1
phone-proxy mypp
    media-termination address 10.10.0.25
    tftp-server address 192.0.2.101 interface inside
    tls-proxy mytls
    ctl-file myctl
    cluster-mode mixed
class-map sec_sccp
    match port tcp 2443
class-map sec_sip
    match port tcp eq 5061
policy-map pp_policy
    class sec_sccp
        inspect skinny phone-proxy mypp
    class sec_sip
        inspect sip phone-proxy mypp
service-policy pp_policy interface outside
```

Example 5: LSC Provisioning in Mixed-mode Cisco UCM cluster; Cisco UCM and TFTP Server on Publisher

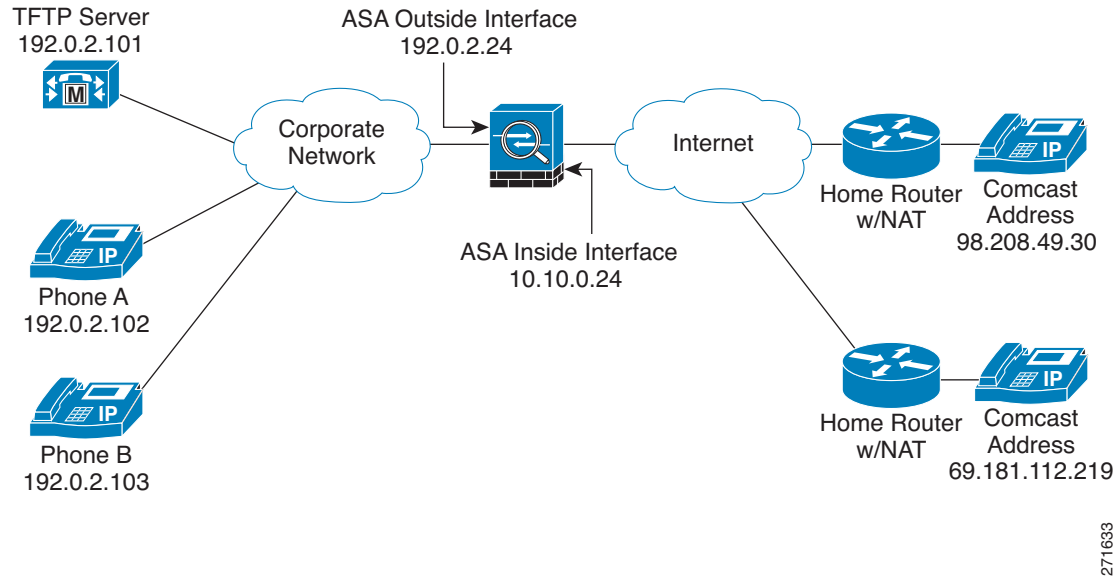
Figure 27-19 shows an example of the configuration for a mixed-mode Cisco UCM cluster where LSC provisioning is required using the following topology.

Note Doing LSC provisioning for remote IP phones is not recommended because it requires that the IP phones first register and they have to register in nonsecure mode. Having the IP phones register in nonsecure mode requires the Administrator to open the nonsecure signaling port for SIP and SCCP on the security appliance. If possible, LSC provisioning should be done inside the corporate network before giving the IP phones to the end-users.

In this sample, you create an access list to allow the IP phones to contact the TFTP server and to allow the IP phones to register in nonsecure mode by opening the nonsecure port for SIP and SCCP as well as the CAPF port for LSC provisioning.

Additionally, you create the CAPF trustpoint by copying and pasting the CAPF certificate from the Cisco UCM Certificate Management software.

Figure 27-19 LSC Provisioning in Mixed-mode Cisco UCM cluster; Cisco UCM and TFTP Server on Publisher




```
static (inside,outside) 10.10.0.26 192.0.2.105
static (inside,outside) udp interface 69 192.0.2.101 69
access-list pp extended permit udp any host 10.10.0.24 eq 69
access-list pp extended permit tcp any host 10.10.0.26 eq 2000
access-list pp extended permit tcp any host 10.10.0.26 eq 5060
access-list pp extended permit tcp any host 10.10.0.26 eq 3804
access-group pp in interface outside
crypto key generate rsa label cluster_kp modulus 1024
crypto ca trustpoint cucm
    enrollment self
    keypair cluster_kp
crypto ca enroll cucm
crypto ca trustpoint tftp_server
    enrollment self
    serial-number
    keypair cluster_kp
crypto ca enroll tftp_server
crypto ca trustpoint capf
    enroll terminal
crypto ca authenticate capf
ctl-file myctl
    record-entry cucm trustpoint cucm_server address 10.10.0.26
    record-entry capf trustpoint capf address 10.10.0.26
    no shutdown
crypto key generate rsa label ldc_signer_key modulus 1024
crypto key generate rsa label phone_common modulus 1024
crypto ca trustpoint ldc_server
    enrollment self
    proxy_ldc_issuer
    fqdn my-ldc-ca.exmaple.com
    subject-name cn=FW_LDC_SIGNER_172_23_45_200
    keypair ldc_signer_key
    crypto ca enroll ldc_server
tls-proxy my_proxy
    server trust-point _internal_PP_myctl
    client ldc issuer ldc_server
    client ldc keypair phone_common
    client cipher-suite aes128-sha1 aes256-sha1
```

```
phone-proxy mypp
    media-termination address 10.10.0.25
    tftp-server address 192.0.2.101 interface inside
    tls-proxy mytls
    ctl-file myctl
    cluster-mode mixed
class-map sec_sccp
    match port tcp 2443
class-map sec_sip
    match port tcp eq 5061
policy-map pp_policy
    class sec_sccp
        inspect skinny phone-proxy mypp
    class sec_sip
        inspect sip phone-proxy mypp
service-policy pp_policy interface outside
```

Example 6: VLAN Transversal

Figure 27-20 shows an example of the configuration to force Cisco IP Communicator (CIPC) softphones to operate in authenticated mode when CIPC softphones are deployed in a voice and data VLAN scenario. VLAN transversal is required between CIPC softphones on the data VLAN and hard phones on the voice VLAN.

In this sample, the Cisco UCM cluster mode is nonsecure.

In this sample, you create an access list to allow the IP phones to contact the TFTP server and to allow the IP phones to register in nonsecure mode by opening the nonsecure port for SIP and SCCP as well as the CAPF port for LSC provisioning.

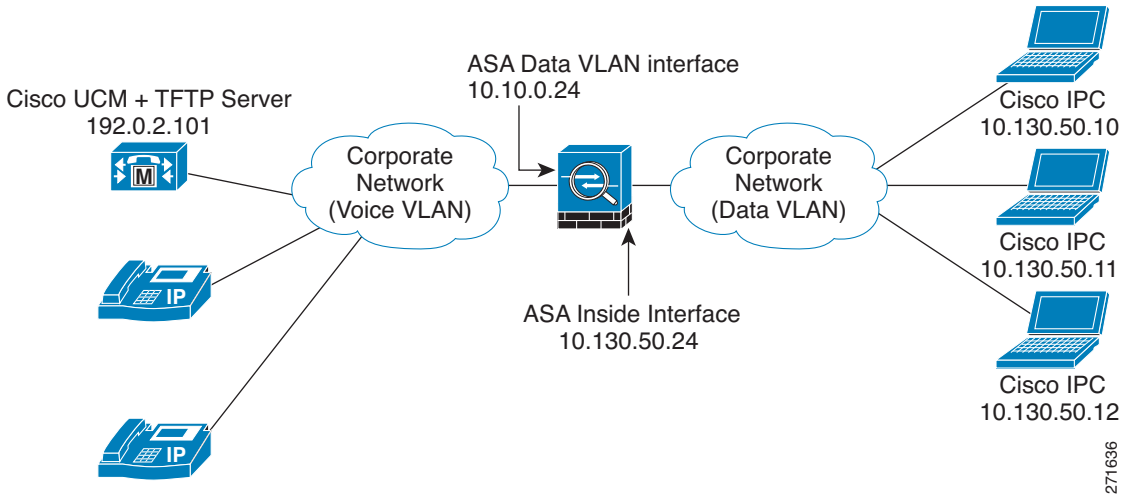
In this sample, you configure NAT for the CIPC by using PAT so that each CIPC is mapped to an IP address space in the Voice VLAN.

Additionally, you create the CAPF trustpoint by copying and pasting the CAPF certificate from the Cisco UCM Certificate Management software.

Note Cisco IP Communicator supports authenticated mode only and does not support encrypted mode; therefore, there is no encrypted voice traffic (SRTP) flowing from the CIPC softphones.

The phone proxy and CIPC are not supported when CIPC is installed on computers in remote locations, such that the calls from those computers traverse the Internet, terminating at the security appliance, to reach IP phones residing on the network behind the adaptive security appliance. The computers where CIPC is installed must be on the network to reach the IP phones behind the security appliance.

Figure 27-20 VLAN Transversal Between CIPC Softphones on the Data VLAN and Hard Phones on the Voice VLAN




```
static (voice,data) 10.130.50.5 192.0.2.101
nat (data) 101 10.130.50.0 255.255.255.0 outside
global (voice) 101 192.0.2.10
access-list pp extended permit udp any host 10.130.50.5 eq 69
access-list pp extended permit tcp any host 10.130.50.5 eq 2000
access-list pp extended permit tcp any host 10.130.50.5 eq 5060
access-list pp extended permit tcp any host 10.130.50.5 eq 3804
access-group pp in interface data
crypto ca generate rsa label cucmtftp_kp modulus 1024
crypto ca trustpoint cucm_tftp_server
    enrollment self
    keypair cucmtftp_kp
crypto ca enroll cucm_tftp_server
crypto ca trustpoint capf
    enrollment terminal
crypto ca authenticate capf
ctl-file myctl
    record-entry cucm-tftp trustpoint cucm_tftp_server address 10.130.50.5
    record-entry capf trustpoint capf address 10.130.50.5
    no shutdown
tls-proxy mytls
    server trust-point _internal_PP_myctl
phone-proxy mypp
    media-termination address 10.130.50.2
    tftp-server address 10.10.0.20 interface inside
    tls-proxy mytls
    ctl-file myctl
    cipc security-mode authenticated
class-map sec_sccp
    match port tcp eq 2443
class-map sec_sip
    match port tcp eq 5061
policy-map pp_policy
    class sec_sccp
        inspect skinny phone-proxy mypp
    class sec_sip
        inspect sip phone-proxy mypp
service-policy pp_policy interface data
```

Cisco Unified Mobility Sample Configurations

This section includes the following topics:

- Example 1: Cisco UMC/Cisco UMA Architecture – Security Appliance as Firewall with TLS Proxy and MMP Inspection, page 27-75
- Example 2: Cisco UMC/Cisco UMA Architecture – Security Appliance as TLS Proxy Only, page 27-76

This section describes sample configurations that apply to two deployment scenarios for the TLS proxy used by the Cisco Unified Mobility solution—scenario 1 where the security appliance functions as both the firewall and TLS proxy and scenario 2 where the security appliance functions as the TLS proxy only. In both scenarios, the clients connect from the Internet.

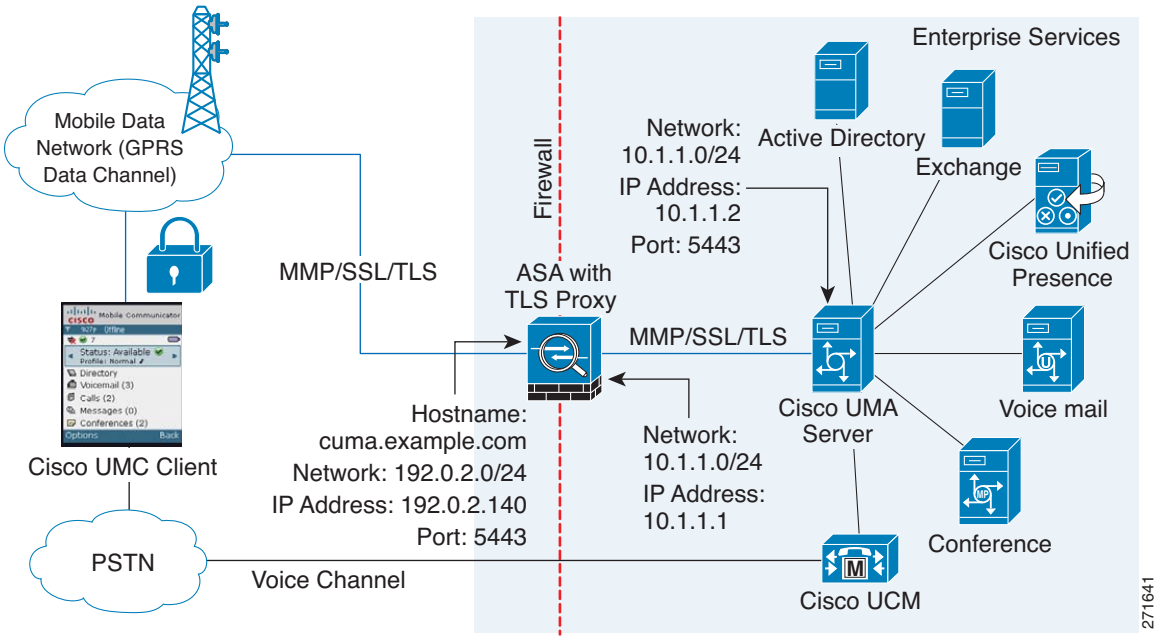
In the samples, you export the Cisco UMA server certificate and key-pair in PKCS-12 format and import it to the security appliance. The certificate will be used during handshake with the Cisco UMA clients.

Installing the Cisco UMA server self-signed certificate in the security appliance truststore is necessary for the security appliance to authenticate the Cisco UMA server during handshake between the security appliance proxy and Cisco UMA server. You create a TLS proxy instance for the Cisco UMA clients connecting to the Cisco UMA server. Lastly, you must enable TLS proxy for MMP inspection.

Example 1: Cisco UMC/Cisco UMA Architecture – Security Appliance as Firewall with TLS Proxy and MMP Inspection

As shown in Figure 27-21 (scenario 1—the recommended architecture), the security appliance functions as both the firewall and TLS proxy. In the scenario 1 deployment, the security appliance is between a Cisco UMA client and a Cisco UMA server. In this scenario, the security appliance performs static NAT by translating the Cisco UMA server 10.1.1.2 IP address to 192.0.2.140.

Figure 27-21 Cisco UMC/Cisco UMA Architecture – Scenario 1: Security Appliance as Firewall with TLS Proxy and MMP Inspection



```
static (inside,outside) 192.0.2.140 10.1.1.2 netmask 255.255.255.255
crypto ca import cuma_proxy pkcs12 sample_passphrase
    <cut-paste base 64 encoded pkcs12 here>
    quit
! for CUMA server's self-signed certificate
crypto ca trustpoint cuma_server
    enrollment terminal
crypto ca authenticate cuma_server
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
MIIDRTCCAu+gAwIBAgIQKVcqP/KW74VP0NZzL+JbRTANBgkqhkiG9w0BAQUFADCB
    [ certificate data omitted ]
/7QEM8izy0EOTSErKu7Nd76jwf5e4qttkQ==
quit
tls-proxy cuma_proxy
    server trust-point cuma_proxy
    no server authenticate-client
    client cipher-suite aes128-sha1 aes256-sha1
class-map cuma_proxy
    match port tcp eq 5443
policy-map global_policy
    class cuma_proxy
        inspect mmp tls-proxy cuma_proxy
service-policy global_policy global
```

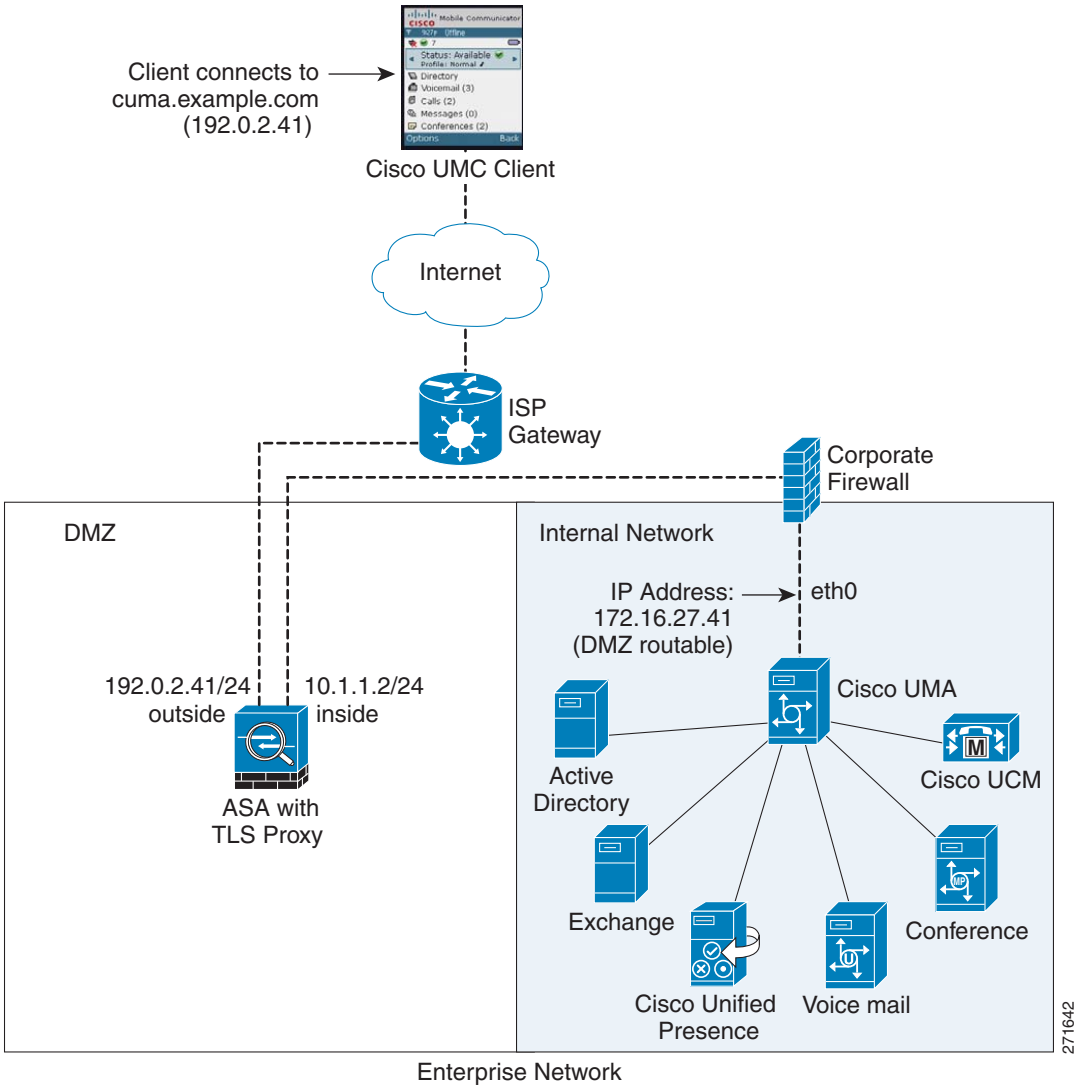
Example 2: Cisco UMC/Cisco UMA Architecture – Security Appliance as TLS Proxy Only

As shown in Figure 27-22 (scenario 2), the security appliance functions as the TLS proxy only and works with an existing firewall. The security appliance and the corporate firewall are performing NAT. The corporate firewall will not be able to predict which client from the Internet needs to connect to the corporate Cisco UMA server. Therefore, to support this deployment, you can take the following actions:

- Set up a NAT rule for inbound traffic that translates the destination IP address 192.0.2.41 to 172.16.27.41.
- Set up an interface PAT rule for inbound traffic translating the source IP address of every packet so that the corporate firewall does not need to open up a wildcard pinhole. The Cisco UMA server receives packets with the source IP address 67.11.12.183.

```
hostname(config)# nat (outside) 1 0.0.0.0 0.0.0.0 outside
hostname(config)# global (inside) 1 10.1.1.2 netmask 255.255.255.255
```

Figure 27-22 Cisco UMC/Cisco UMA Architecture – Scenario 2: Security Appliance as TLS Proxy Only



Enterprise Network

```
static (inside,outside) 192.0.2.41 172.16.27.41 netmask 255.255.255.255
nat (outside) 1 0.0.0.0 0.0.0.0 outside
global (inside) 1 10.1.1.2 netmask 255.255.255.255
crypto ca import cuma_proxy pkcs12 sample_passphrase
    <cut-paste base 64 encoded pkcs12 here>
    quit
! for CUMA server's self-signed certificate
crypto ca trustpoint cuma_server
    enrollment terminal
crypto ca authenticate cuma_server
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
MIIDRTCCAu+gAwIBAgIQKVcqP/KW74VP0NZzL+JbRTANBgkqhkiG9w0BAQUFADCB
    [ certificate data omitted ]
/7QEM8izy0EOTSErKu7Nd76jwf5e4qttkQ==
quit
tls-proxy cuma_proxy
    server trust-point cuma_proxy
    no server authenticate-client
```

```
    client cipher-suite aes128-sha1 aes256-sha1
class-map cuma_proxy
    match port tcp eq 5443
policy-map global_policy
    class cuma_proxy
        inspect mmp tls-proxy cuma_proxy
service-policy global_policy global
```

Cisco Unified Presence Sample Configuration

The following sample illustrates the necessary configuration for the security appliance to perform TLS proxy for Cisco Unified Presence as shown in Figure 27-23. It is assumed that a single Cisco UP (Entity X) is in the local domain and self-signed certificates are used between Entity X and the ASA.

For each Cisco UP that could initiate a connection (by sending SIP SUBSCRIBE) to the foreign server, you must also configure static PAT and if you have another Cisco UP with the address (10.0.0.3 in this sample), it must use a different set of PAT ports (such as 45062 or 45070). Dynamic NAT or PAT can be used for outbound connections or TLS handshake. The security appliance SIP inspection engine takes care of the necessary translation (fixup).

When you create the necessary RSA key pairs, a key pair is used by the self-signed certificate presented to Entity X (proxy for Entity Y). When you create a proxy certificate for Entity Y, the certificate is installed on the Entity X truststore. It could also be enrolled with a local CA trusted by Entity X.

Exporting the security appliance self-signed certificate (ent_y_proxy) and installing it as a trusted certificate on Entity X is necessary for Entity X to authenticate the security appliance. Exporting the Entity X certificate and installing it on the security appliance is needed for the security appliance to authenticate Entity X during handshake with X. If Entity X uses a self-signed certificate, the self-signed certificate must be installed; if Entity X uses a CA issued the certificate, the CA's certificated needs to be installed.

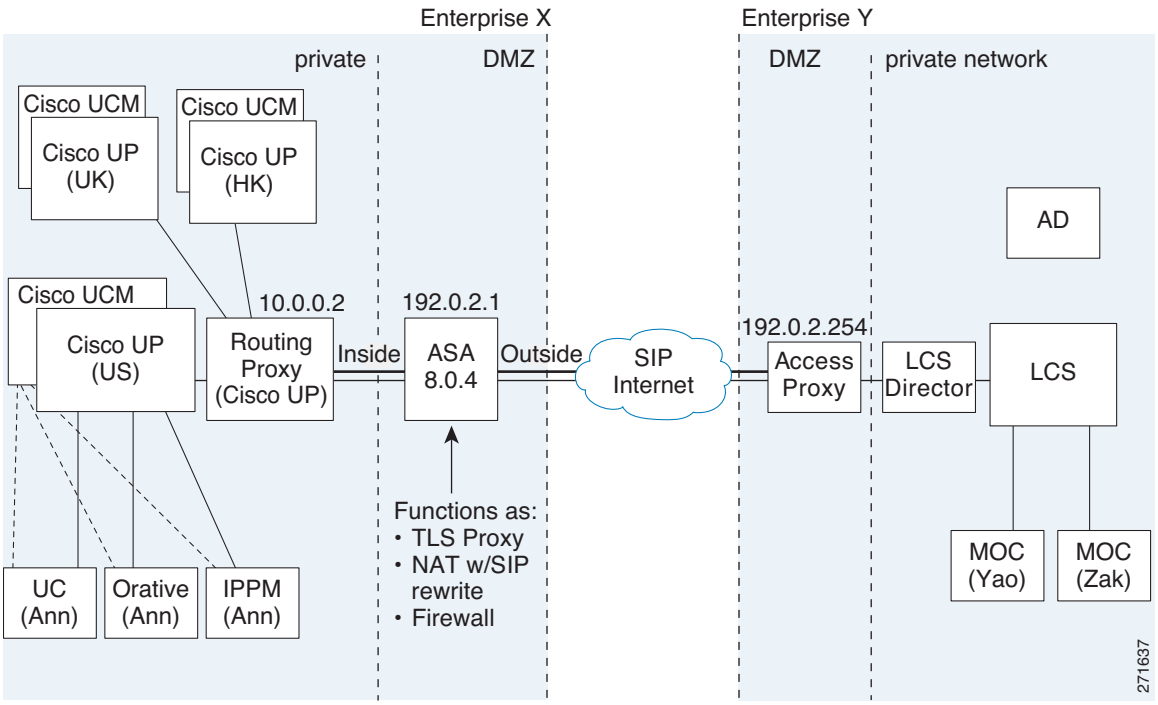
For about obtaining a certificate from a trusted CA, see Certificate Configuration, page 41-5.

Installing the CA certificate that signs the Entity Y certificate on the security appliance is necessary for the security appliance to authenticate Entity Y.

When creating TLS proxy instances for Entity X and Entity Y, the entity that initiates the TLS connection is in the role of "TLS client". Because the TLS proxy has strict definition of "client" and "server" proxy, two TLS proxy instances must be defined if either of the entities could initiate the connection.

When enabling the TLS proxy for SIP inspection, policies must be defined for both entities that could initiate the connection.

Figure 27-23 Typical Cisco Unified Presence/LCS Federation Scenario




```
static (inside,outside) tcp 192.0.2.1 5061 10.0.0.2 5061 netmask 255.255.255.255
static (inside,outside) tcp 192.0.2.1 5062 10.0.0.2 5062 netmask 255.255.255.255
static (inside,outside) udp 192.0.2.1 5070 10.0.0.2 5070 netmask 255.255.255.255
static (inside,outside) tcp 192.0.2.1 45062 10.0.0.3 5062 netmask 255.255.255.255
static (inside,outside) udp 192.0.2.1 45070 10.0.0.3 5070 netmask 255.255.255.255
global (outside) 102 192.0.2.1 netmask 255.255.255.255
nat (inside) 102 0.0.0.0 0.0.0.0
crypto key generate rsa label ent_y_proxy_key modulus 1024
    ! for self-signed Entity Y proxy certificate
crypto ca trustpoint ent_y_proxy
    enrollment self
    fqdn none
    subject-name cn=Ent-Y-Proxy
    keypair ent_y_proxy_key
crypto ca enroll ent_y_proxy
crypto ca export ent_y_proxy identity-certificate
! for Entity X's self-signed certificate
crypto ca trustpoint ent_x_cert
    enrollment terminal
crypto ca authenticate ent_x_cert
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
    [ certificate data omitted ]
quit
! for Entity Y's CA certificate
crypto ca trustpoint ent_y_ca
    enrollment terminal
crypto ca authenticate ent_y_ca
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
MIIDRTCCAu+gAwIBAgIQKVcqP/KW74VP0NZzL+JbRTANBgkqhkiG9w0BAQUFADCB
    [ certificate data omitted ]
/7QEM8izy0EOTSErKu7Nd76jwf5e4qttkQ==
quit
```

```
! Entity X to Entity Y
tls-proxy ent_x_to_y
    server trust-point ent_y_proxy
    client trust-point ent_x_proxy
    client cipher-suite aes128-sha1 aes256-sha1 3des-sha1 null-sha1
! Entity Y to Entity X
tls-proxy ent_y_to_x
    server trust-point ent_x_proxy
    client trust-point ent_y_proxy
    client cipher-suite aes128-sha1 aes256-sha1 3des-sha1 null-sha1
access-list ent_x_to_y extended permit tcp host 10.0.0.2 host 192.0.2.254 eq 5061
access-list ent_y_to_x extended permit tcp host 192.0.2.254 host 192.0.2.1 eq 5061
class-map ent_x_to_y
    match access-list ent_x_to_y
class-map ent_y_to_x
    match access-list ent_y_to_x
policy-map type inspect sip sip_inspect
    parameters
        ! SIP inspection parameters
policy-map global_policy
    class ent_x_to_y
        inspect sip sip_inspect tls-proxy ent_x_to_y
    class ent_y_to_x
        inspect sip sip_inspect tls-proxy ent_y_to_x
service-policy global_policy global
```



CHAPTER 28

Configuring ARP Inspection and Bridging Parameters for Transparent Mode

This chapter describes how to enable ARP inspection and how to customize bridging operations for the security appliance in transparent mode. In multiple context mode, the commands in this chapter can be entered in a security context, but not the system.

This chapter includes the following sections:

- Configuring ARP Inspection, page 28-1
- Customizing the MAC Address Table, page 28-3

Configuring ARP Inspection

This section describes ARP inspection and how to enable it, and includes the following topics:

- ARP Inspection Overview, page 28-1
- Adding a Static ARP Entry, page 28-2
- Enabling ARP Inspection, page 28-2

ARP Inspection Overview

By default, all ARP packets are allowed through the security appliance. You can control the flow of ARP packets by enabling ARP inspection.

When you enable ARP inspection, the security appliance compares the MAC address, IP address, and source interface in all ARP packets to static entries in the ARP table, and takes the following actions:

- If the IP address, MAC address, and source interface match an ARP entry, the packet is passed through.
- If there is a mismatch between the MAC address, the IP address, or the interface, then the security appliance drops the packet.
- If the ARP packet does not match any entries in the static ARP table, then you can set the security appliance to either forward the packet out all interfaces (flood), or to drop the packet.

Note The dedicated management interface, if present, never floods packets even if this parameter is set to flood.

ARP inspection prevents malicious users from impersonating other hosts or routers (known as ARP spoofing). ARP spoofing can enable a "man-in-the-middle" attack. For example, a host sends an ARP request to the gateway router; the gateway router responds with the gateway router MAC address. The attacker, however, sends another ARP response to the host with the attacker MAC address instead of the router MAC address. The attacker can now intercept all the host traffic before forwarding it on to the router.

ARP inspection ensures that an attacker cannot send an ARP response with the attacker MAC address, so long as the correct MAC address and the associated IP address are in the static ARP table.

Adding a Static ARP Entry

ARP inspection compares ARP packets with static ARP entries in the ARP table. Although hosts identify a packet destination by an IP address, the actual delivery of the packet on Ethernet relies on the Ethernet MAC address. When a router or host wants to deliver a packet on a directly connected network, it sends an ARP request asking for the MAC address associated with the IP address, and then delivers the packet to the MAC address according to the ARP response. The host or router keeps an ARP table so it does not have to send ARP requests for every packet it needs to deliver. The ARP table is dynamically updated whenever ARP responses are sent on the network, and if an entry is not used for a period of time, it times out. If an entry is incorrect (for example, the MAC address changes for a given IP address), the entry times out before it can be updated.

Note The transparent firewall uses dynamic ARP entries in the ARP table for traffic to and from the security appliance, such as management traffic.

To add a static ARP entry, enter the following command:

```
hostname(config)# arp interface_name ip_address mac_address
```

For example, to allow ARP responses from the router at 10.1.1.1 with the MAC address 0009.7cbe.2100 on the outside interface, enter the following command:

```
hostname(config)# arp outside 10.1.1.1 0009.7cbe.2100
```

Enabling ARP Inspection

To enable ARP inspection, enter the following command:

```
hostname(config)# arp-inspection interface_name enable [flood | no-flood]
```

Where **flood** forwards non-matching ARP packets out all interfaces, and **no-flood** drops non-matching packets.

Note The default setting is to flood non-matching packets. To restrict ARP through the security appliance to only static entries, then set this command to **no-flood**.

For example, to enable ARP inspection on the outside interface, and to drop all non-matching ARP packets, enter the following command:

```
hostname(config)# arp-inspection outside enable no-flood
```

To view the current settings for ARP inspection on all interfaces, enter the **show arp-inspection** command.

Customizing the MAC Address Table

This section describes the MAC address table, and includes the following topics:

- MAC Address Table Overview, page 28-3
- Adding a Static MAC Address, page 28-3
- Setting the MAC Address Timeout, page 28-4
- Disabling MAC Address Learning, page 28-4
- Viewing the MAC Address Table, page 28-4

MAC Address Table Overview

The security appliance learns and builds a MAC address table in a similar way as a normal bridge or switch: when a device sends a packet through the security appliance, the security appliance adds the MAC address to its table. The table associates the MAC address with the source interface so that the security appliance knows to send any packets addressed to the device out the correct interface.

The ASA 5505 adaptive security appliance includes a built-in switch; the switch MAC address table maintains the MAC address-to-switch port mapping for traffic within each VLAN. This section discusses the bridge MAC address table, which maintains the MAC address-to-VLAN interface mapping for traffic that passes between VLANs.

Because the security appliance is a firewall, if the destination MAC address of a packet is not in the table, the security appliance does not flood the original packet on all interfaces as a normal bridge does. Instead, it generates the following packets for directly connected devices or for remote devices:

- Packets for directly connected devices—The security appliance generates an ARP request for the destination IP address, so that the security appliance can learn which interface receives the ARP response.
- Packets for remote devices—The security appliance generates a ping to the destination IP address so that the security appliance can learn which interface receives the ping reply.

The original packet is dropped.

Adding a Static MAC Address

Normally, MAC addresses are added to the MAC address table dynamically as traffic from a particular MAC address enters an interface. You can add static MAC addresses to the MAC address table if desired. One benefit to adding static entries is to guard against MAC spoofing. If a client with the same MAC address as a static entry attempts to send traffic to an interface that does not match the static entry, then the security appliance drops the traffic and generates a system message. When you add a static ARP entry (see the "Adding a Static ARP Entry" section on page 28-2), a static MAC address entry is automatically added to the MAC address table.

To add a static MAC address to the MAC address table, enter the following command:

```
hostname(config)# mac-address-table static interface_name mac_address
```

The *interface_name* is the source interface.

Setting the MAC Address Timeout

The default timeout value for dynamic MAC address table entries is 5 minutes, but you can change the timeout. To change the timeout, enter the following command:

```
hostname(config)# mac-address-table aging-time timeout_value
```

The *timeout_value* (in minutes) is between 5 and 720 (12 hours). 5 minutes is the default.

Disabling MAC Address Learning

By default, each interface automatically learns the MAC addresses of entering traffic, and the security appliance adds corresponding entries to the MAC address table. You can disable MAC address learning if desired, however, unless you statically add MAC addresses to the table, no traffic can pass through the security appliance.

To disable MAC address learning, enter the following command:

```
hostname(config)# mac-learn interface_name disable
```

The **no** form of this command reenables MAC address learning. The **clear configure mac-learn** command reenables MAC address learning on all interfaces.

Viewing the MAC Address Table

You can view the entire MAC address table (including static and dynamic entries for both interfaces), or you can view the MAC address table for an interface. To view the MAC address table, enter the following command:

```
hostname# show mac-address-table [interface_name]
```

The following is sample output from the **show mac-address-table** command that shows the entire table:

```
hostname# show mac-address-table
interface          mac address       type      Time Left
------------------------------------------------------------------------
outside            0009.7cbe.2100    static      -
inside             0010.7cbe.6101    static      -
inside             0009.7cbe.5101    dynamic     10
```

The following is sample output from the **show mac-address-table** command that shows the table for the inside interface:

```
hostname# show mac-address-table inside
interface          mac address       type      Time Left
------------------------------------------------------------------------
inside             0010.7cbe.6101    static      -
inside             0009.7cbe.5101    dynamic     10
```



PART 3

Configuring VPN



CHAPTER 29

Configuring IPsec and ISAKMP

This chapter describes how to configure the IPsec and ISAKMP standards to build Virtual Private Networks. It includes the following sections:

- Tunneling Overview, page 29-1
- IPsec Overview, page 29-2
- Configuring ISAKMP, page 29-2
- Configuring Certificate Group Matching, page 29-9
- Configuring IPsec, page 29-11
- Clearing Security Associations, page 29-26
- Clearing Crypto Map Configurations, page 29-27
- Supporting the Nokia VPN Client, page 29-27

Tunneling Overview

Tunneling makes it possible to use a public TCP/IP network, such as the Internet, to create secure connections between remote users and a private corporate network. Each secure connection is called a tunnel.

The security appliance uses the ISAKMP and IPsec tunneling standards to build and manage tunnels. ISAKMP and IPsec accomplish the following:

- Negotiate tunnel parameters
- Establish tunnels
- Authenticate users and data
- Manage security keys
- Encrypt and decrypt data
- Manage data transfer across the tunnel
- Manage data transfer inbound and outbound as a tunnel endpoint or router

The security appliance functions as a bidirectional tunnel endpoint. It can receive plain packets from the private network, encapsulate them, create a tunnel, and send them to the other end of the tunnel where they are unencapsulated and sent to their final destination. It can also receive encapsulated packets from the public network, unencapsulate them, and send them to their final destination on the private network.

IPsec Overview

The security appliance uses IPsec for LAN-to-LAN VPN connections, and provides the option of using IPsec for client-to-LAN VPN connections. In IPsec terminology, a *peer* is a remote-access client or another secure gateway. For both connection types, the security appliance supports only Cisco peers. Because we adhere to VPN industry standards, ASAs may work with other vendors' peers; however, we do not support them.

During tunnel establishment, the two peers negotiate security associations that govern authentication, encryption, encapsulation, and key management. These negotiations involve two phases: first, to establish the tunnel (the IKE SA); and second, to govern traffic within the tunnel (the IPsec SA).

A LAN-to-LAN VPN connects networks in different geographic locations. In IPsec LAN-to-LAN connections, the security appliance can function as initiator or responder. In IPsec client-to-LAN connections, the security appliance functions only as responder. Initiators propose SAs; responders accept, reject, or make counter-proposals—all in accordance with configured SA parameters. To establish a connection, both entities must agree on the SAs.

Note When the security appliance is configured for IPsec VPN, you cannot enable security contexts (also called firewall multmode) or Active/Active stateful failover. Therefore, these features are unavailable.

Configuring ISAKMP

This section describes the Internet Key Exchange protocol which is also called the Internet Security Association and Key Management Protocol. The security appliance IKE commands use ISAKMP as a keyword, which this guide echoes. ISAKMP works with IPsec to make VPNs more scalable. This section includes the following topics:

- ISAKMP Overview, page 29-2
- Configuring ISAKMP Policies, page 29-5
- Enabling ISAKMP on the Outside Interface, page 29-6
- Disabling ISAKMP in Aggressive Mode, page 29-6
- Determining an ID Method for ISAKMP Peers, page 29-6
- Enabling IPsec over NAT-T, page 29-7
- Enabling IPsec over TCP, page 29-8
- Waiting for Active Sessions to Terminate Before Rebooting, page 29-8
- Alerting Peers Before Disconnecting, page 29-9

ISAKMP Overview

IKE, also called ISAKMP, is the negotiation protocol that lets two hosts agree on how to build an IPsec security association. ISAKMP separates negotiation into two phases: Phase 1 and Phase 2.

Phase 1 creates the first tunnel, which protects later ISAKMP negotiation messages. Phase 2 creates the tunnel that protects data.

To set the terms of the ISAKMP negotiations, you create an ISAKMP policy, which includes the following:

- An authentication method, to ensure the identity of the peers.
- An encryption method, to protect the data and ensure privacy.
- A Hashed Message Authentication Codes (HMAC) method to ensure the identity of the sender, and to ensure that the message has not been modified in transit.
- A Diffie-Hellman group to determine the strength of the encryption-key-determination algorithm. The security appliance uses this algorithm to derive the encryption and hash keys.
- A limit to the time the security appliance uses an encryption key before replacing it.

Table 29-1 provides information about the ISAKMP policy keywords and their values.

Table 29-1 ISAKMP Policy Keywords for CLI Commands

Command	Keyword	Meaning	Description
crypto isakmp policy authentication	rsa-sig	A digital certificate with keys generated by the RSA signatures algorithm	Specifies the authentication method the security appliance uses to establish the identity of each IPsec peer.
	crack	Challenge/Response for Authenticated Cryptographic Keys	CRACK provides strong mutual authentication when the client authenticates using a legacy method such as RADIUS and the server uses public key authentication.
	pre-share (default)	Preshared keys	Preshared keys do not scale well with a growing network but are easier to set up in a small network.
crypto isakmp policy encryption	des	56-bit DES-CBC	Specifies the symmetric encryption algorithm that protects data transmitted between two IPsec peers. The default is 168-bit Triple DES.
	3des (default)	168-bit Triple DES	
	aes aes-192 aes-256		The Advanced Encryption Standard supports key lengths of 128, 192, 256 bits.
crypto isakmp policy hash	sha (default)	SHA-1 (HMAC variant)	Specifies the hash algorithm used to ensure data integrity. It ensures that a packet comes from where it says it comes from, and that it has not been modified in transit.
	md5	MD5 (HMAC variant)	The default is SHA-1. MD5 has a smaller digest and is considered to be slightly faster than SHA-1. A successful (but extremely difficult) attack against MD5 has occurred; however, the HMAC variant IKE uses prevents this attack.

Table 29-1 ISAKMP Policy Keywords for CLI Commands (continued)

Command	Keyword	Meaning	Description
crypto isakmp policy group	1 2 (default) 5	Group 1 (768-bit) Group 2 (1024-bit) Group 5 (1536-bit)	Specifies the Diffie-Hellman group identifier, which the two IPsec peers use to derive a shared secret without transmitting it to each other. The lower the Diffie-Hellman group number, the less CPU time it requires to execute. The higher the Diffie-Hellman group no., the greater the security. Cisco VPN Client Version 3.x or higher requires a minimum of Group 2. (If you configure DH Group 1, the Cisco VPN Client cannot connect.) AES support is available on security appliances licensed for VPN-3DES only. To support the large key sizes required by AES, ISAKMP negotiation should use Diffie-Hellman (DH) Group 5.
crypto isakmp policy lifetime	integer value (86400 = default)	120 to 2147483647 seconds	Specifies the SA lifetime. The default is 86,400 seconds or 24 hours. As a general rule, a shorter lifetime provides more secure ISAKMP negotiations (up to a point). However, with shorter lifetimes, the security appliance sets up future IPsec SAs more quickly.

Each configuration supports a maximum of 20 ISAKMP policies, each with a different set of values. Assign a unique priority to each policy you create. The lower the priority number, the higher the priority.

When ISAKMP negotiations begin, the peer that initiates the negotiation sends all of its policies to the remote peer, and the remote peer tries to find a match. The remote peer checks all of the peer's policies against each of its configured policies in priority order (highest priority first) until it discovers a match.

A match exists when both policies from the two peers contain the same encryption, hash, authentication, and Diffie-Hellman parameter values, and when the remote peer policy specifies a lifetime less than or equal to the lifetime in the policy the initiator sent. If the lifetimes are not identical, the security appliance uses the shorter lifetime. If no acceptable match exists, ISAKMP refuses negotiation and the SA is not established.

There is an implicit trade-off between security and performance when you choose a specific value for each parameter. The level of security the default values provide is adequate for the security requirements of most organizations. If you are interoperating with a peer that supports only one of the values for a parameter, your choice is limited to that value.



Note New ASA configurations do not have a default ISAKMP policy.

Configuring ISAKMP Policies

To configure ISAKMP policies, in global configuration mode, use the **crypto isakmp policy** command with its various arguments. The syntax for ISAKMP policy commands is as follows:

crypto isakmp policy *priority* **attribute_name** [**attribute_value |** *integer*]

You must include the priority in each of the ISAKMP commands. The priority number uniquely identifies the policy, and determines the priority of the policy in ISAKMP negotiations.

To enable and configure ISAKMP, complete the following steps, using the examples as a guide:

Note If you do not specify a value for a given policy parameter, the default value applies.

Step 1 Specify the encryption algorithm. The default is Triple DES. This example sets encryption to DES.

crypto isakmp policy *priority* **encryption [aes | aes-192 | aes-256 | des** | **3des]**

For example:

```
hostname(config)# crypto isakmp policy 2 encryption des
```

Step 2 Specify the hash algorithm. The default is SHA-1. This example configures MD5.

crypto isakmp policy *priority* **hash [md5** | **sha]**

For example:

```
hostname(config)# crypto isakmp policy 2 hash md5
```

Step 3 Specify the authentication method. The default is preshared keys. This example configures RSA signatures.

crypto isakmp policy *priority* **authentication [pre-share | crack | rsa-sig]**

For example:

```
hostname(config)# crypto isakmp policy 2 authentication rsa-sig
```

Step 4 Specify the Diffie-Hellman group identifier. The default is Group 2. This example configures Group 5.

crypto isakmp policy *priority* **group** [**1** | **2 | 5**]

For example:

```
hostname(config)# crypto isakmp policy 2 group  5
```

Step 5 Specify the SA lifetime. This examples sets a lifetime of 4 hours (14400 seconds). The default is 86400 seconds (24 hours).

crypto isakmp policy *priority* **lifetime** *seconds*

For example:

```
hostname(config)# crypto isakmp policy 2 lifetime 14400
```

Enabling ISAKMP on the Outside Interface

You must enable ISAKMP on the interface that terminates the VPN tunnel. Typically this is the outside, or public interface.

To enable ISAKMP, enter the following command:

crypto isakmp enable *interface-name*

For example:

```
hostname(config)# crypto isakmp enable outside
```

Disabling ISAKMP in Aggressive Mode

Phase 1 ISAKMP negotiations can use either main mode or aggressive mode. Both provide the same services, but aggressive mode requires only two exchanges between the peers totaling 3 messages, rather than three exchanges totaling 6 messages. Aggressive mode is faster, but does not provide identity protection for the communicating parties. Therefore, the peers must exchange identification information prior to establishing a secure SA. Aggressive mode is enabled by default.

- Main mode is slower, using more exchanges, but it protects the identities of the communicating peers.
- Aggressive mode is faster, but does not protect the identities of the peers.

To disable ISAKMP in aggressive mode, enter the following command:

```
crypto isakmp am-disable
```

For example:

```
hostname(config)# crypto isakmp am-disable
```

If you have disabled aggressive mode, and want to revert to back to it, use the **no** form of the command. For example:

```
hostname(config)# no crypto isakmp am-disable
```

Note Disabling aggressive mode prevents Cisco VPN clients from using preshared key authentication to establish tunnels to the security appliance. However, they may use certificate-based authentication (that is, ASA or RSA) to establish tunnels.

Determining an ID Method for ISAKMP Peers

During Phase I ISAKMP negotiations the peers must identify themselves to each other. You can choose the identification method from the following options:

Address	Uses the IP addresses of the hosts exchanging ISAKMP identity information.
Automatic	Determines ISAKMP negotiation by connection type: • IP address for preshared key. • Cert Distinguished Name for certificate authentication.
Hostname	Uses the fully qualified domain name of the hosts exchanging ISAKMP identity information (default). This name comprises the hostname and the domain name.
Key ID	Uses the string the remote peer uses to look up the preshared key.

The security appliance uses the Phase I ID to send to the peer. This is true for all VPN scenarios except LAN-to-LAN connections in main mode that authenticate with preshared keys.

The default setting is hostname.

To change the peer identification method, enter the following command:

crypto isakmp identity {address | **hostname** | **key-id** ***id-string*** | **auto}**

For example, the following command sets the peer identification method to automatic:

```
hostname(config)# crypto isakmp identity auto
```

Enabling IPsec over NAT-T

NAT-T lets IPsec peers establish a connection through a NAT device. It does this by encapsulating IPsec traffic in UDP datagrams, using port 4500, thereby providing NAT devices with port information. NAT-T auto-detects any NAT devices, and only encapsulates IPsec traffic when necessary. This feature is disabled by default.

With the exception of the home zone on the Cisco ASA 5505, the security appliance can simultaneously support standard IPsec, IPsec over TCP, NAT-T, and IPsec over UDP, depending on the client with which it is exchanging data. When both NAT-T and IPsec over UDP are enabled, NAT-T takes precedence. IPsec over TCP, if enabled, takes precedence over all other connection methods.

When you enable NAT-T, the security appliance automatically opens port 4500 on all IPsec enabled interfaces.

The security appliance supports multiple IPsec peers behind a single NAT/PAT device operating in one of the following networks, but not both:

- LAN-to-LAN
- Remote access

In a mixed environment, the remote access tunnels fail the negotiation because all peers appear to be coming from the same public IP address, that of the NAT device. Also, remote access tunnels fail in a mixed environment because they often use the same name as the LAN-to-LAN tunnel group (that is, the IP address of the NAT device). This match can cause negotiation failures among multiple peers in a mixed LAN-to-LAN and remote access network of peers behind the NAT device.

Using NAT-T

To use NAT-T, you must perform the following tasks:

Step 1 Enter the following command to enable IPsec over NAT-T globally on the security appliance.

crypto isakmp nat-traversal *natkeepalive*

natkeepalive is in the range 10 to 3600 seconds. The default is 20 seconds.

For example, enter the following command to enable NAT-T and set the keepalive to one hour.

```
hostname(config)# crypto isakmp nat-traversal 3600
```

Step 2 Select the “before-fragmentation” option for the IPsec fragmentation policy.

This option lets traffic travel across NAT devices that do not support IP fragmentation. It does not impede the operation of NAT devices that do support IP fragmentation.

Enabling IPsec over TCP

IPsec over TCP enables a Cisco VPN client to operate in an environment in which standard ESP or ISAKMP cannot function, or can function only with modification to existing firewall rules. IPsec over TCP encapsulates both the ISAKMP and IPsec protocols within a TCP-like packet, and enables secure tunneling through both NAT and PAT devices and firewalls. This feature is disabled by default.



Note This feature does not work with proxy-based firewalls.

IPsec over TCP works with remote access clients. You enable it globally, and it works on all ISAKMP enabled interfaces. It is a client to security appliance feature only. It does not work for LAN-to-LAN connections.

The security appliance can simultaneously support standard IPsec, IPsec over TCP, NAT-Traversal, and IPsec over UDP, depending on the client with which it is exchanging data. IPsec over TCP, if enabled, takes precedence over all other connection methods.

The VPN 3002 hardware client, which supports one tunnel at a time, can connect using standard IPsec, IPsec over TCP, NAT-Traversal, or IPsec over UDP.

You enable IPsec over TCP on both the security appliance and the client to which it connects.

You can enable IPsec over TCP for up to 10 ports that you specify. If you enter a well-known port, for example port 80 (HTTP) or port 443 (HTTPS), the system displays a warning that the protocol associated with that port no longer works on the public interface. The consequence is that you can no longer use a browser to manage the security appliance through the public interface. To solve this problem, reconfigure the HTTP/HTTPS management to different ports.

The default port is 10000.

You must configure TCP port(s) on the client as well as on the security appliance. The client configuration must include at least one of the ports you set for the security appliance.

To enable IPsec over TCP globally on the security appliance, enter the following command:

crypto isakmp ipsec-over-tcp [port *port 1...port0***]**

This example enables IPsec over TCP on port 45:

```
hostname(config)# crypto isakmp ctcp port 45
```

Waiting for Active Sessions to Terminate Before Rebooting

You can schedule a security appliance reboot to occur only when all active sessions have terminated voluntarily. This feature is disabled by default.

To enable waiting for all active sessions to voluntarily terminate before the security appliance reboots, enter the following command:

crypto isakmp reload-wait

For example:

```
hostname(config)# crypto isakmp reload-wait
```

Use the **reload** command to reboot the security appliance. If you set the **reload-wait** command, you can use the **reload quick** command to override the **reload-wait** setting. The **reload** and **reload-wait** commands are available in privileged EXEC mode; neither includes the **isakmp** prefix.

Alerting Peers Before Disconnecting

Remote access or LAN-to-LAN sessions can drop for several reasons, such as: a security appliance shutdown or reboot, session idle timeout, maximum connection time exceeded, or administrator cut-off.

The security appliance can notify qualified peers (in LAN-to-LAN configurations), Cisco VPN clients and VPN 3002 hardware clients of sessions that are about to be disconnected. The peer or client receiving the alert decodes the reason and displays it in the event log or in a pop-up pane. This feature is disabled by default.

Qualified clients and peers include the following:

- Security appliances with Alerts enabled.
- Cisco VPN clients running version 4.0 or later software (no configuration required).
- VPN 3002 hardware clients running version 4.0 or later software, and with Alerts enabled.
- VPN 3000 series concentrators running version 4.0 or later software, with Alerts enabled.

To enable disconnect notification to IPsec peers, enter the **crypto isakmp disconnect-notify** command.

For example:

```
hostname(config)# crypto isakmp disconnect-notify
```

Configuring Certificate Group Matching

Tunnel groups define user connection terms and permissions. Certificate group matching lets you match a user to a tunnel group using either the Subject DN or Issuer DN of the user certificate.

To match users to tunnel groups based on these fields of the certificate, you must first create rules that define a matching criteria, and then associate each rule with the desired tunnel group.

To create a certificate map, **use the crypto ca certificate map** command. To define a tunnel group, use the **tunnel-group** command.

You must also configure a certificate group matching policy that sets one of the following methods for identifying the permission groups of certificate users:

- Match the group from the rules
- Match the group from the organizational unit (OU) field
- Use a default group for all certificate users

You can use any or all of these methods.

Creating a Certificate Group Matching Rule and Policy

To configure the policy and rules by which certificate-based ISAKMP sessions map to tunnel groups, and to associate the certificate map entries with tunnel groups, enter the **tunnel-group-map** command in global configuration mode.

The syntax follows:

tunnel-group-map enable {*rules* | *ou* | *ike-id* | *peer ip*}

tunnel-group-map [*rule-index*] **enable** *policy*

policy	Specifies the policy for deriving the tunnel group name from the certificate. *Policy* can be one of the following: *ike-id*—Indicates that if a tunnel-group is not determined based on a rule lookup or taken from the ou, then the certificate-based ISAKMP sessions are mapped to a tunnel group based on the content of the phase1 ISAKMP ID. *ou*—Indicates that if a tunnel-group is not determined based on a rule lookup, then use the value of the OU in the subject distinguished name (DN). *peer-ip*—Indicates that if a tunnel-group is not determined based on a rule lookup or taken from the ou or ike-id methods, then use the peer IP address. *rules*—Indicates that the certificate-based ISAKMP sessions are mapped to a tunnel group based on the certificate map associations configured by this command.
rule index	(Optional) Refers to parameters specified by the **crypto ca certificate map** command. The values are 1 to 65535.

Be aware of the following:

- You can invoke this command multiple times as long as each invocation is unique and you do not reference a map index more than once.
- Rules cannot be longer than 255 characters.
- You can assign multiple rules to the same group. To do that, you add the rule priority and group first. Then you define as many criteria statements as you need for each group. When multiple rules are assigned to the same group, a match results for the first rule that tests true.
- Create a single rule if you want to require all criteria to match before assigning a user to a specific tunnel group. Requiring all criteria to match is equivalent to a logical AND operation. Alternatively, create one rule for each criterion if you want to require that only one match before assigning a user to a specific tunnel group. Requiring only one criterion to match is equivalent to a logical OR operation.

The following example enables mapping of certificate-based ISAKMP sessions to a tunnel group based on the content of the phase1 ISAKMP ID:

```
hostname(config)# tunnel-group-map enable ike-id
hostname(config)#
```

The following example enables mapping of certificate-based ISAKMP sessions to a tunnel group based on the IP address of the peer:

```
hostname(config)# tunnel-group-map enable peer-ip
hostname(config)#
```

The following example enables mapping of certificate-based ISAKMP sessions based on the organizational unit (OU) in the subject distinguished name (DN):

```
hostname(config)# tunnel-group-map enable ou
hostname(config)#
```

The following example enables mapping of certificate-based ISAKMP sessions based on established rules:

```
hostname(config)# tunnel-group-map enable rules
hostname(config)#
```

Using the Tunnel-group-map default-group Command

This command specifies a default tunnel group to use when the configuration does not specify a tunnel group.

The syntax is **tunnel-group-map** [*rule-index*] **default-group** *tunnel-group-name* where the *rule-index* is the priority for the rule, and *tunnel-group name* must be for a tunnel group that already exists.

Configuring IPsec

This section provides background information about IPsec and describes the procedures required to configure the security appliance when using IPsec to implement a VPN. It contains the following topics:

- Understanding IPsec Tunnels, page 29-11
- Understanding Transform Sets, page 29-12
- Defining Crypto Maps, page 29-12
- Applying Crypto Maps to Interfaces, page 29-19
- Using Interface Access Lists, page 29-19
- Changing IPsec SA Lifetimes, page 29-21
- Creating a Basic IPsec Configuration, page 29-22
- Using Dynamic Crypto Maps, page 29-23
- Providing Site-to-Site Redundancy, page 29-26
- Viewing an IPsec Configuration, page 29-26

Understanding IPsec Tunnels

IPsec tunnels are sets of SAs that the security appliance establishes between peers. The SAs define the protocols and algorithms to apply to sensitive data, and also specify the keying material the peers use. IPsec SAs control the actual transmission of user traffic. SAs are unidirectional, but are generally established in pairs (inbound and outbound).

The peers negotiate the settings to use for each SA. Each SA consists of the following:

- Transform sets
- Crypto maps
- Access lists
- Tunnel groups
- Prefragmentation policies

Understanding Transform Sets

A transform set is a combination of security protocols and algorithms that define how the security appliance protects data. During IPsec SA negotiations, the peers must identify a transform set that is the same at both peers. The security appliance then applies the matching transform set to create an SA that protects data flows in the access list for that crypto map.

The security appliance tears down the tunnel if you change the definition of the transform set used to create its SA. See "Clearing Security Associations" for further information.

Note If you clear or delete the only element in a transform set, the security appliance automatically removes the crypto map references to it.

Defining Crypto Maps

Crypto maps define the IPsec policy to be negotiated in the IPsec SA. They include the following:

- Access list to identify the packets that the IPsec connection permits and protects.
- Peer identification
- Local address for the IPsec traffic (See "Applying Crypto Maps to Interfaces" for more details.)
- Up to six transform sets with which to attempt to match the peer security settings.

A *crypto map set* consists of one or more crypto maps that have the same map name. You create a crypto map set when you create its first crypto map. The following command syntax creates or adds to a crypto map:

```
crypto map map-name seq-num match address access-list-name
```

You can continue to enter this command to add crypto maps to the crypto map set. In the following example, "mymap" is the name of the crypto map set to which you might want to add crypto maps:

```
crypto map mymap 10 match address 101
```

The *sequence number* (seq-num) shown in the syntax above distinguishes one crypto map from another one with the same name. The sequence number assigned to a crypto map also determines its priority among the other crypto maps within a crypto map set. The lower the sequence number, the higher the priority. After you assign a crypto map set to an interface, the security appliance evaluates all IP traffic passing through the interface against the crypto maps in the set, beginning with the crypto map with the lowest sequence number.

The ACL assigned to a crypto map consists of all of the ACEs that have the same access-list-name, as shown in the following command syntax:

```
access-list access-list-name {deny | permit} ip source source-netmask destination
destination-netmask
```

Each ACL consists of one or more ACEs that have the same access-list-name. You create an ACL when you create its first ACE. The following command syntax creates or adds to an ACL:

```
access-list access-list-name {deny | permit} ip source source-netmask destination
destination-netmask
```

In the following example, the security appliance applies the IPsec protections assigned to the crypto map to all traffic flowing from the 10.0.0.0 subnet to the 10.1.1.0 subnet.

```
access-list 101 permit ip 10.0.0.0 255.255.255.0 10.1.1.0 255.255.255.0
```

The crypto map that matches the packet determines the security settings used in the SA negotiations. If the local security appliance initiates the negotiation, it uses the policy specified in the static crypto map to create the offer to send to the specified peer. If the peer initiates the negotiation, the security appliance attempts to match the policy to a static crypto map, and if that fails, any dynamic crypto maps in the crypto map set, to decide whether to accept or reject the peer offer.

For two peers to succeed in establishing an SA, they must have at least one compatible crypto map. To be compatible, a crypto map must meet the following criteria:

- The crypto map must contain compatible crypto ACLs (for example, mirror image ACLs). If the responding peer uses dynamic crypto maps, so must the security appliance as a requirement to apply IPsec.
- Each crypto map identifies the other peer (unless the responding peer uses dynamic crypto maps).
- The crypto maps have at least one transform set in common.

You can apply only one crypto map set to a single interface. Create more than one crypto map for a particular interface on the security appliance if any of the following conditions exist:

- You want specific peers to handle different data flows.
- You want different IPsec security to apply to different types of traffic.

For example, create a crypto map and assign an ACL to identify traffic between two subnets and assign one transform set. Create another crypto map with a different ACL to identify traffic between another two subnets and apply a transform set with different VPN parameters.

If you create more than one crypto map for an interface, specify a sequence number (seq-num) for each map entry to determine its priority within the crypto map set.

Each ACE contains a permit or deny statement. Table 29-2 explains the special meanings of permit and deny ACEs in ACLs applied to crypto maps.

Table 29-2 Special Meanings of Permit and Deny in Crypto Access Lists Applied to Outbound Traffic

Result of Crypto Map Evaluation	Response
Match criterion in an ACE containing a permit statement	Halt further evaluation of the packet against the remaining ACEs in the crypto map set, and evaluate the packet security settings against those in the transform sets assigned to the crypto map. After matching the security settings to those in a transform set, the security appliance applies the associated IPsec settings. Typically for outbound traffic, this means that it decrypts, authenticates, and routes the packet.
Match criterion in an ACE containing a deny statement	Interrupt further evaluation of the packet against the remaining ACEs in the crypto map under evaluation, and resume evaluation against the ACEs in the next crypto map, as determined by the next seq-num assigned to it.
Fail to match all tested permit ACEs in the crypto map set	Route the packet without encrypting it.

ACEs containing deny statements filter out outbound traffic that does not require IPsec protection (for example, routing protocol traffic). Therefore, insert initial deny statements to filter outbound traffic that should not be evaluated against permit statements in a crypto access list.

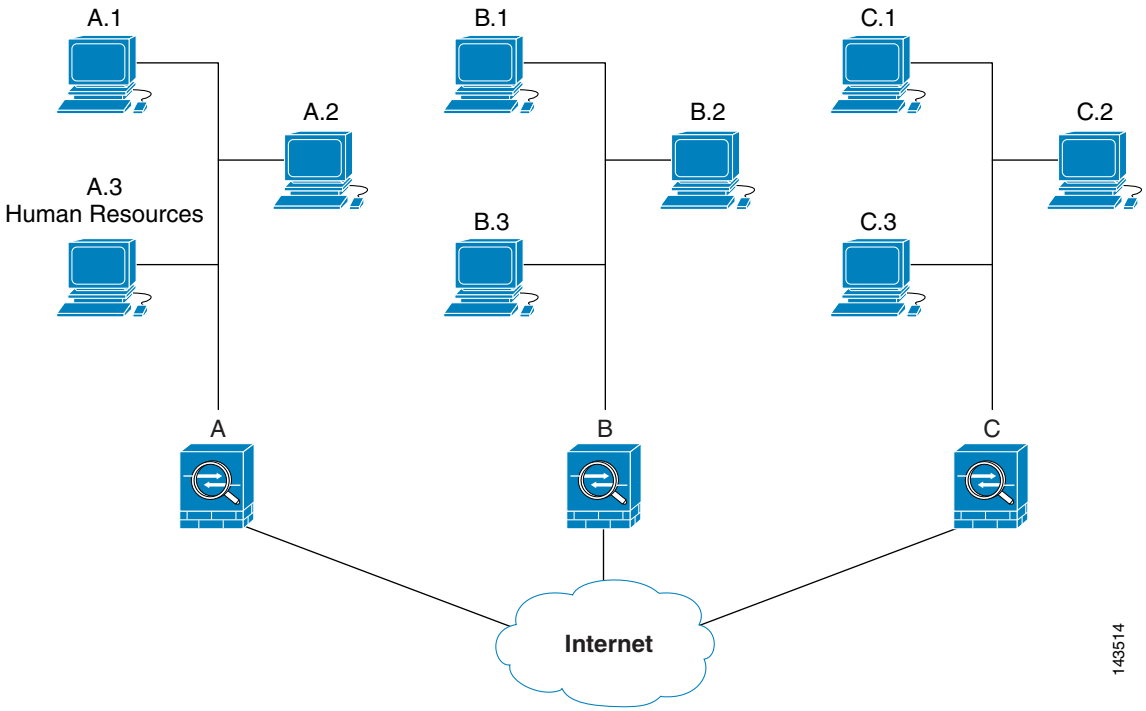
For an inbound, encrypted packet, the security appliance uses the source address and ESP SPI to determine the decryption parameters. After the security appliance decrypts the packet, it compares the inner header of the decrypted packet to the permit ACEs in the ACL associated with the packet SA. If the inner header fails to match the proxy, the security appliance drops the packet. It the inner header matches the proxy, the security appliance routes the packet.

When comparing the inner header of an inbound packet that was not encrypted, the security appliance ignores all deny rules because they would prevent the establishment of a Phase 2 SA.

Note To route inbound, unencrypted traffic as clear text, insert deny ACEs before permit ACEs.

Figure 29-1 shows an example LAN-to-LAN network of security appliances.

Figure 29-1 Effect of Permit and Deny ACEs on Traffic (Conceptual Addresses)



The simple address notation shown in this figure and used in the following explanation is an abstraction. An example with real IP addresses follows the explanation.

The objective in configuring Security Appliances A, B, and C in this example LAN-to-LAN network is to permit tunneling of all traffic originating from one of the hosts shown in Figure 29-1 and destined for one of the other hosts. However, because traffic from Host A.3 contains sensitive data from the Human Resources department, it requires strong encryption and more frequent rekeying than the other traffic. So we want to assign a special transform set for traffic from Host A.3.

To configure Security Appliance A for outbound traffic, we create two crypto maps, one for traffic from Host A.3 and the other for traffic from the other hosts in Network A, as shown in the following example:

```
Crypto Map Seq_No_1
    deny packets from A.3 to B
    deny packets from A.3 to C
```

```
    permit packets from A to B
    permit packets from A to C
Crypto Map Seq_No_2
    permit packets from A.3 to B
    permit packets from A.3 to C
```

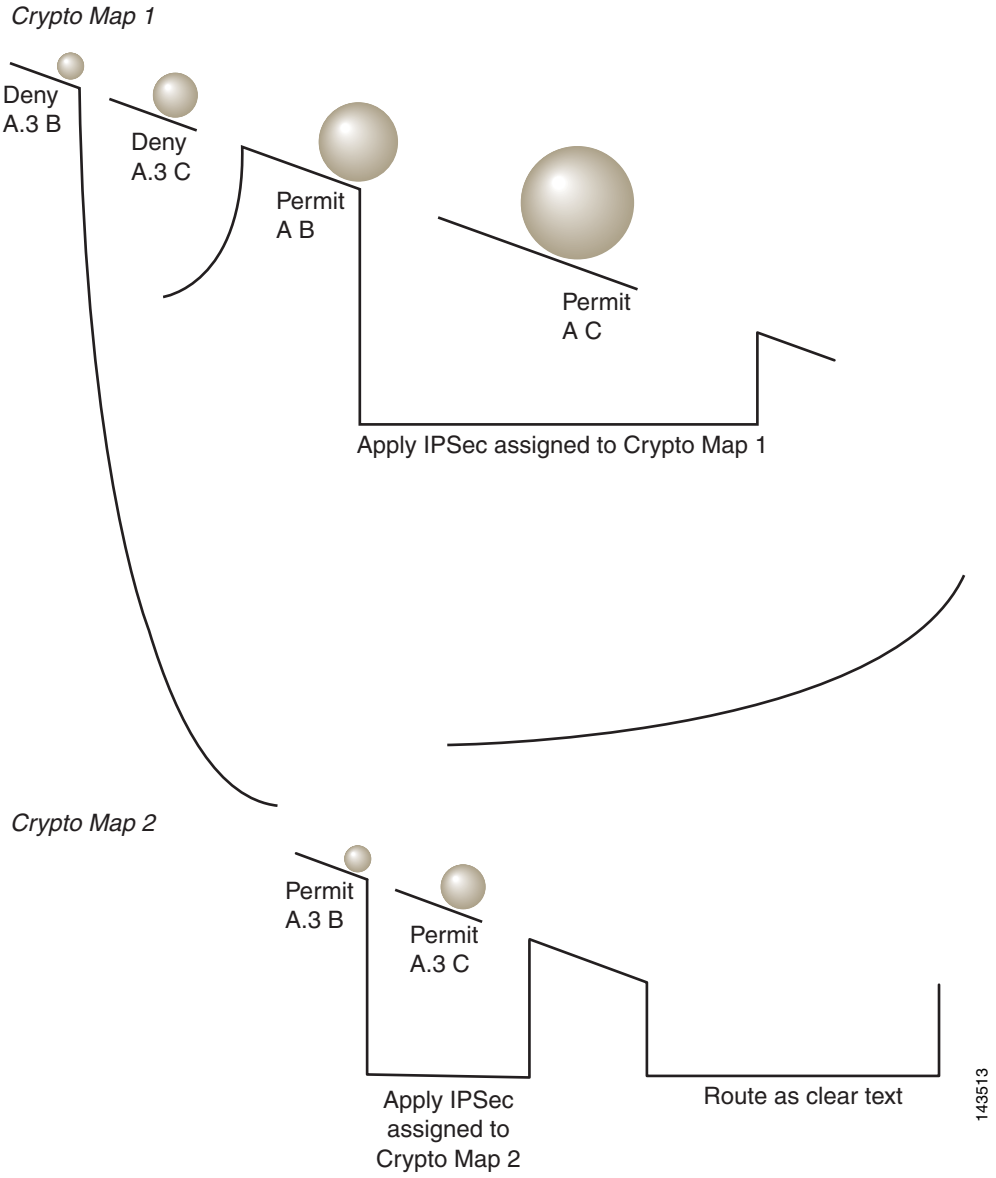
After creating the ACLs, you assign a transform set to each crypto map to apply the required IPsec to each matching packet.

Cascading ACLs involves the insertion of deny ACEs to bypass evaluation against an ACL and resume evaluation against a subsequent ACL in the crypto map set. Because you can associate each crypto map with different IPsec settings, you can use deny ACEs to exclude special traffic from further evaluation in the corresponding crypto map, and match the special traffic to permit statements in another crypto map to provide or require different security. The sequence number assigned to the crypto ACL determines its position in the evaluation sequence within the crypto map set.

Figure 29-2 shows the cascading ACLs created from the conceptual ACEs above. The meaning of each symbol in the figure follows.

	Crypto map within a crypto map set.
	(Gap in a straight line) Exit from a crypto map when a packet matches an ACE.
	Packet that fits the description of one ACE. Each size ball represents a different packet matching the respective ACE in the figure. The differences in size merely represent differences in the source and destination of each packet.
	Redirection to the next crypto map in the crypto map set.
	Response when a packet either matches an ACE or fails to match all of the permit ACEs in a crypto map set.

Figure 29-2 Cascading ACLs in a Crypto Map Set



Security Appliance A evaluates a packet originating from Host A.3 until it matches a permit ACE and attempts to assign the IPsec security associated with the crypto map. Whenever the packet matches a deny ACE, the security appliance ignores the remaining ACEs in the crypto map and resumes evaluation against the next crypto map, as determined by the sequence number assigned to it. So in the example, if Security Appliance A receives a packet from Host A.3, it matches the packet to a deny ACE in the first crypto map and resumes evaluation of the packet against the next crypto map. When it matches the packet to the permit ACE in that crypto map, it applies the associated IPsec security (strong encryption and frequent rekeying).

To complete the security appliance configuration in the example network, we assign mirror crypto maps to Security Appliances B and C. However, because security appliances ignore deny ACEs when evaluating inbound, encrypted traffic, we can omit the mirror equivalents of the deny A.3 B and deny A.3 C ACEs, and therefore omit the mirror equivalents of Crypto Map 2. So the configuration of cascading ACLs in Security Appliances B and C is unnecessary.

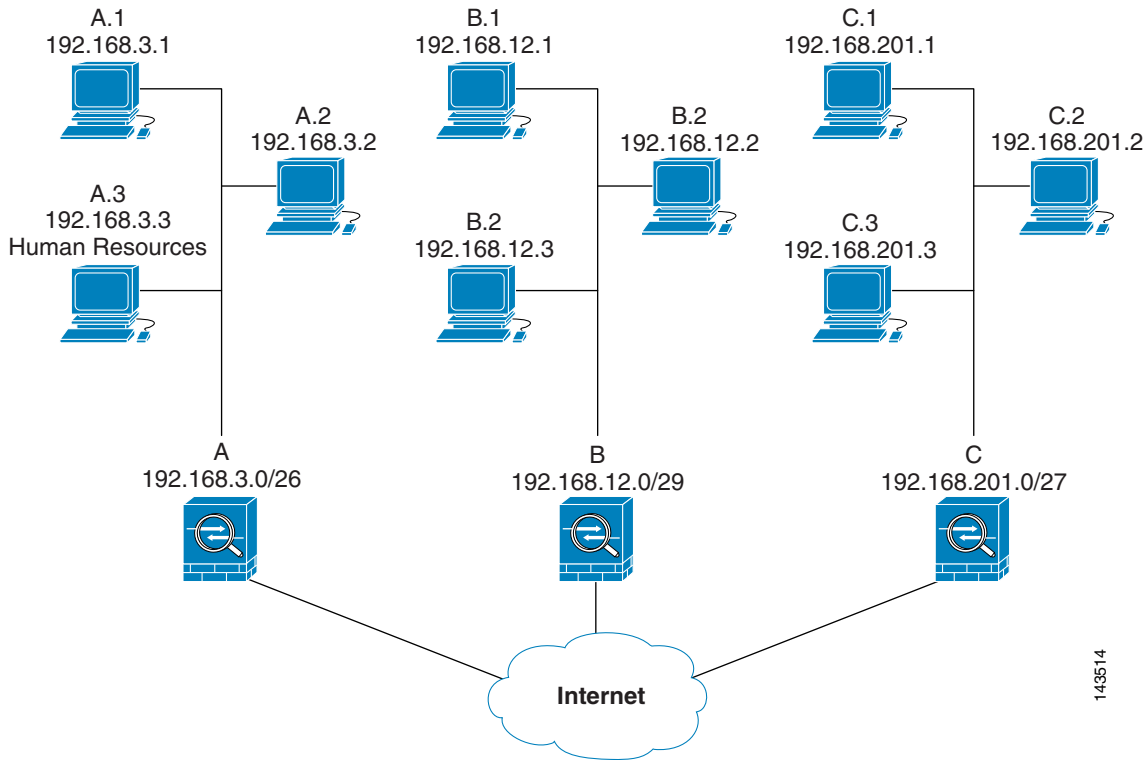
Table 29-3 shows the ACLs assigned to the crypto maps configured for all three security appliances in Figure 29-1.

Table 29-3 Example Permit and Deny Statements (Conceptual)

Security Appliance A		**Security Appliance B**		**Security Appliance C**	
Crypto Map Sequence No.	**ACE Pattern**	**Crypto Map Sequence No.**	**ACE Pattern**	**Crypto Map Sequence No.**	**ACE Pattern**
1	deny A.3 B	1	permit B A	1	permit C A
	deny A.3 C				
	permit A B				
	permit A C		permit B C		permit C B
2	permit A.3 B				
	permit A.3 C				

Figure 29-3 maps the conceptual addresses shown in Figure 29-1 to real IP addresses.

Figure 29-3 Effect of Permit and Deny ACEs on Traffic (Real Addresses)



The tables that follow combine the IP addresses shown in Figure 29-3 to the concepts shown in Table 29-3. The real ACEs shown in these tables ensure that all IPsec packets under evaluation within this network receive the proper IPsec settings.

Table 29-4 Example Permit and Deny Statements for Security Appliance A

Security Appliance	Crypto Map Sequence No.	ACE Pattern	Real ACEs
A	1	deny A.3 B	deny 192.168.3.3 255.255.255.192 192.168.12.0 255.255.255.248
		deny A.3 C	deny 192.168.3.3 255.255.255.192 192.168.201.0 255.255.255.224
		permit A B	permit 192.168.3.0 255.255.255.192 192.168.12.0 255.255.255.248
		permit A C	permit 192.168.3.0 255.255.255.192 192.168.201.0 255.255.255.224
	2	permit A.3 B	permit 192.168.3.3 255.255.255.192 192.168.12.0 255.255.255.248
		permit A.3 C	permit 192.168.3.3 255.255.255.192 192.168.201.0 255.255.255.224
B	None needed	permit B A	permit 192.168.12.0 255.255.255.248 192.168.3.0 255.255.255.192
		permit B C	permit 192.168.12.0 255.255.255.248 192.168.201.0 255.255.255.224
C	None needed	permit C A	permit 192.168.201.0 255.255.255.224 192.168.3.0 255.255.255.192
		permit C B	permit 192.168.201.0 255.255.255.224 192.168.12.0 255.255.255.248

You can apply the same reasoning shown in the example network to use cascading ACLs to assign different security settings to different hosts or subnets protected by a Cisco security appliance.

Note By default, the security appliance does not support IPsec traffic destined for the same interface from which it enters. (Names for this type of traffic include U-turn, hub-and-spoke, and hairpinning.) However, you might want IPsec to support U-turn traffic. To do so, insert an ACE to permit traffic to and from the network. For example, to support U-turn traffic on Security Appliance B, add a conceptual "permit B B" ACE to ACL1. The actual ACE would be as follows:

```
permit 192.168.12.0 255.255.255.248 192.168.12.0 255.255.255.248
```

Applying Crypto Maps to Interfaces

You must assign a crypto map set to each interface through which IPsec traffic flows. The security appliance supports IPsec on all interfaces. Assigning the crypto map set to an interface instructs the security appliance to evaluate all the traffic against the crypto map set and to use the specified policy during connection or SA negotiation.

Assigning a crypto map to an interface also initializes run-time data structures, such as the SA database and the security policy database. Reassigning a modified crypto map to the interface resynchronizes the run-time data structures with the crypto map configuration. Also, adding new peers through the use of new sequence numbers and reassigning the crypto map does not tear down existing connections.

Using Interface Access Lists

By default, the security appliance lets IPsec packets bypass interface ACLs. If you want to apply interface access lists to IPsec traffic, use the **no** form of the **sysopt connection permit-vpn** command.

The crypto map access list bound to the outgoing interface either permits or denies IPsec packets through the VPN tunnel. IPsec authenticates and deciphers packets that arrive from an IPsec tunnel, and subjects them to evaluation against the ACL associated with the tunnel.

Access lists define which IP traffic to protect. For example, you can create access lists to protect all IP traffic between two subnets or two hosts. (These access lists are similar to access lists used with the **access-group** command. However, with the **access-group** command, the access list determines which traffic to forward or block at an interface.)

Before the assignment to crypto maps, the access lists are not specific to IPsec. Each crypto map references the access lists and determines the IPsec properties to apply to a packet if it matches a permit in one of the access lists.

Access lists assigned to IPsec crypto maps have four primary functions:

- Select outbound traffic to be protected by IPsec (permit = protect).
- Trigger an ISAKMP negotiation for data travelling without an established SA.
- Process inbound traffic to filter out and discard traffic that should have been protected by IPsec.
- Determine whether to accept requests for IPsec SAs when processing IKE negotiation from the peer. (Negotiation applies only to **ipsec-isakmp crypto map** entries.) The peer must "permit" a data flow associated with an **ipsec-isakmp crypto map** command entry to ensure acceptance during negotiation.

Regardless of whether the traffic is inbound or outbound, the security appliance evaluates traffic against the access lists assigned to an interface. You assign IPsec to an interface as follows:

Step 1 Create the access lists to be used for IPsec.

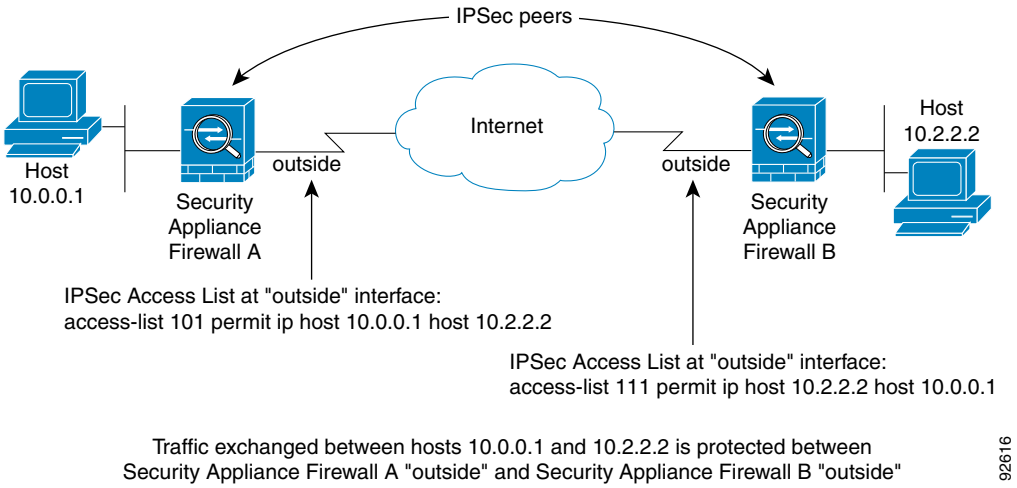
Step 2 Map the lists to one or more crypto maps, using the same crypto map name.

Step 3 Map the transform sets to the crypto maps to apply IPsec to the data flows.

Step 4 Apply the crypto maps collectively as a "crypto map set" by assigning the crypto map name they share to the interface.

In Figure 29-4, IPsec protection applies to traffic between Host 10.0.0.1 and Host 10.2.2.2 as the data exits the outside interface on Security Appliance A toward Host 10.2.2.2.

Figure 29-4 How Crypto Access Lists Apply to IPsec



Security Appliance A evaluates traffic from Host 10.0.0.1 to Host 10.2.2.2, as follows:

- source = host 10.0.0.1
- dest = host 10.2.2.2

Security Appliance A also evaluates traffic from Host 10.2.2.2 to Host 10.0.0.1, as follows:

- source = host 10.2.2.2
- dest = host 10.0.0.1

The first permit statement that matches the packet under evaluation determines the scope of the IPsec SA.

Note If you delete the only element in an access list, the security appliance also removes the associated crypto map.

If you modify an access list currently referenced by one or more crypto maps, use the **crypto map interface** command to reinitialize the run-time SA database. See the **crypto map** command for more information.

We recommend that for every crypto access list specified for a static crypto map that you define at the local peer, you define a "mirror image" crypto access list at the remote peer. The crypto maps should also support common transforms and refer to the other system as a peer. This ensures correct processing of IPsec by both peers.

Note Every static crypto map must define an access list and an IPsec peer. If either is missing, the crypto map is incomplete and the security appliance drops any traffic that it has not already matched to an earlier, complete crypto map. Use the **show conf** command to ensure that every crypto map is complete. To fix an incomplete crypto map, remove the crypto map, add the missing entries, and reapply it.

We discourage the use of the **any** keyword to specify source or destination addresses in crypto access lists because they cause problems. We strongly discourage the **permit any any** command statement because it does the following:

- Protects all outbound traffic, including all protected traffic sent to the peer specified in the corresponding crypto map.
- Requires protection for all inbound traffic.

  In this scenario, the security appliance silently drops all inbound packets that lack IPsec protection.

Be sure that you define which packets to protect. If you use the **any** keyword in a **permit** statement, preface it with a series of **deny** statements to filter out traffic that would otherwise fall within that **permit** statement that you do not want to protect.

Note Decrypted "through" traffic is permitted from the client despite having an access-group on the outside interface, which calls a "deny ip any any" access-list, while **no sysopt connection permit-vpn** is configured.

Users who want to control access to the protected network via Site-to-Site or remote access VPN using the **no sysopt permit** command in conjunction with an access control list (ACL) on the outside interface are not successful.

In this situation, when management-access inside is enabled, the ACL is not applied, and users can still connect using SSH to the security appliance. Traffic to hosts on the inside network are blocked correctly by the ACL, but can't block decrypted "through" traffic to the inside interface.

The **ssh** and **http** commands are of a higher priority than the ACLs. In other words, to deny ssh, telnet, or ICMP traffic to the box from the VPN session, use ssh, telnet and icmp commands, which denies the IP local pool should be added.

Changing IPsec SA Lifetimes

You can change the global lifetime values that the security appliance uses when negotiating new IPsec SAs. You can override these global lifetime values for a particular crypto map.

IPsec SAs use a derived, shared, secret key. The key is an integral part of the SA; they time out together to require the key to refresh. Each SA has two lifetimes: “timed” and “traffic-volume.” An SA expires after the respective lifetime and negotiations begin for a new one. The default lifetimes are 28,800 seconds (eight hours) and 4,608,000 kilobytes (10 megabytes per second for one hour).

If you change a global lifetime, the security appliance drops the tunnel. It uses the new value in the negotiation of subsequently established SAs.

When a crypto map does not have configured lifetime values and the security appliance requests a new SA, it inserts the global lifetime values used in the existing SA into the request sent to the peer. When a peer receives a negotiation request, it uses the smaller of either the lifetime value the peer proposes or the locally configured lifetime value as the lifetime of the new SA.

The peers negotiate a new SA before crossing the lifetime threshold of the existing SA to ensure that a new SA is ready when the existing one expires. The peers negotiate a new SA when about 5 to 15 percent of the lifetime of the existing SA remains.

Creating a Basic IPsec Configuration

You can create basic IPsec configurations with static or dynamic crypto maps.

To create a basic IPsec configuration using a static crypto map, perform the following steps:

Step 1 To create an access list to define the traffic to protect, enter the following command:

access-list *access-list-name* **{deny | permit} ip** *source source-netmask destination destination-netmask*

For example:

```
access-list 101 permit ip 10.0.0.0 255.255.255.0 10.1.1.0 255.255.255.0
```

In this example, the **permit** keyword causes all traffic that matches the specified conditions to be protected by crypto.

Step 2 To configure a transform set that defines how to protect the traffic, enter the following command:

crypto ipsec transform-set *transform-set-name transform1* [*tcansform2, transform3*]

For example:

```
crypto ipsec transform-set myset1 esp-des esp-sha-hmac
crypto ipsec transform-set myset2 esp-3des esp-sha-hmac
crypto ipsec transform-set aes_set esp-md5-hmac esp-aes-256
```

In this example, “myset1” and “myset2” and “aes_set” are the names of the transform sets.

Step 3 To create a crypto map, perform the following steps:

a. Assign an access list to a crypto map:

crypto map *map-name seq-num* **match address** *access-list-name*

In the following example, “mymap” is the name of the crypto map set. The map set sequence number 10, which is used to rank multiple entries within one crypto map set. The lower the sequence number, the higher the priority.

```
crypto map mymap 10 match address 101
```

In this example, the access list named 101 is assigned to crypto map “mymap.”

b. Specify the peer to which the IPsec protected traffic can be forwarded:

crypto map *map-name seq-num* **set peer** *ip-address*

For example:

```
crypto map mymap 10 set peer 192.168.1.100
```

The security appliance sets up an SA with the peer assigned the IP address 192.168.1.100. Specify multiple peers by repeating this command.

c. Specify which transform sets are allowed for this crypto map. List multiple transform sets in order of priority (highest priority first). You can specify up to 11 transform sets in a crypto map.

```
crypto map map-name seq-num set transform-set transform-set-name1
[transform-set-name2, …transform-set-name6]
```

For example:

```
crypto map mymap 10 set transform-set myset1 myset2
```

In this example, when traffic matches access list 101, the SA can use either "myset1" (first priority) or "myset2" (second priority) depending on which transform set matches the transform set of the peer.

d. (Optional) Specify an SA lifetime for the crypto map if you want to override the global lifetime.

```
crypto map map-name seq-num set security-association lifetime {seconds seconds |
kilobytes kilobytes}
```

For example:

```
crypto map mymap 10 set security-association lifetime seconds 2700
```

This example shortens the timed lifetime for the crypto map "mymap 10" to 2700 seconds (45 minutes). The traffic volume lifetime is not changed.

e. (Optional) Specify that IPsec require perfect forward secrecy when requesting new SA for this crypto map, or require PFS in requests received from the peer:

```
crypto map map-name seq-num set pfs [group1 | group2 | group5]
```

For example:

```
crypto map mymap 10 set pfs group2
```

This example requires PFS when negotiating a new SA for the crypto map "mymap 10." The security appliance uses the 1024-bit Diffie-Hellman prime modulus group in the new SA.

Step 4 Apply a crypto map set to an interface for evaluating IPsec traffic:

```
crypto map map-name interface interface-name
```

For example:

```
crypto map mymap interface outside
```

In this example, the security appliance evaluates the traffic going through the outside interface against the crypto map "mymap" to determine whether it needs to be protected.

Using Dynamic Crypto Maps

A dynamic crypto map is a crypto map without all of the parameters configured. It acts as a policy template where the missing parameters are later dynamically learned, as the result of an IPsec negotiation, to match the peer requirements. The security appliance applies a dynamic crypto map to let a peer negotiate a tunnel if its IP address is not already identified in a static crypto map. This occurs with the following types of peers:

- Peers with dynamically assigned public IP addresses.

  Both LAN-to-LAN and remote access peers can use DHCP to obtain a public IP address. The security appliance uses this address only to initiate the tunnel.

- Peers with dynamically assigned private IP addresses.

Peers requesting remote access tunnels typically have private IP addresses assigned by the headend. Generally, LAN-to-LAN tunnels have a predetermined set of private networks that are used to configure static maps and therefore used to establish IPsec SAs.

As an administrator configuring static crypto maps, you might not know the IP addresses that are dynamically assigned (via DHCP or some other method), and you might not know the private IP addresses of other clients, regardless of how they were assigned. VPN clients typically do not have static IP addresses; they require a dynamic crypto map to allow IPsec negotiation to occur. For example, the headend assigns the IP address to a Cisco VPN client during IKE negotiation, which the client then uses to negotiate IPsec SAs.

Note A dynamic crypto map requires only the **transform-set** parameter.

Dynamic crypto maps can ease IPsec configuration and we recommend them for use in networks where the peers are not always predetermined. Use dynamic crypto maps for Cisco VPN clients (such as mobile users) and routers that obtain dynamically assigned IP addresses.

Tip Use care when using the **any** keyword in **permit** entries in dynamic crypto maps. If the traffic covered by such a **permit** entry could include multicast or broadcast traffic, insert **deny** entries for the appropriate address range into the access list. Remember to insert **deny** entries for network and subnet broadcast traffic, and for any other traffic that IPsec should not protect.

Dynamic crypto maps work only to negotiate SAs with remote peers that initiate the connection. The security appliance cannot use dynamic crypto maps to initiate connections to a remote peer. With a dynamic crypto map, if outbound traffic matches a permit entry in an access list and the corresponding SA does not yet exist, the security appliance drops the traffic.

A crypto map set may include a dynamic crypto map. Dynamic crypto map sets should be the lowest priority crypto maps in the crypto map set (that is, they should have the highest sequence numbers) so that the security appliance evaluates other crypto maps first. It examines the dynamic crypto map set only when the other (static) map entries do not match.

Similar to static crypto map sets, a dynamic crypto map set consists of all of the dynamic crypto maps with the same dynamic-map-name. The dynamic-seq-num differentiates the dynamic crypto maps in a set. If you configure a dynamic crypto map, insert a permit ACL to identify the data flow of the IPsec peer for the crypto access list. Otherwise the security appliance accepts any data flow identity the peer proposes.

Caution Do not assign static (default) routes for traffic to be tunneled to a security appliance interface configured with a dynamic crypto map set. To identify the traffic that should be tunneled, add the ACLs to the dynamic crypto map. Use care to identify the proper address pools when configuring the ACLs associated with remote access tunnels. Use Reverse Route Injection to install routes only after the tunnel is up.

The procedure for using a dynamic crypto map entry is the same as the basic configuration described in "Creating a Basic IPsec Configuration," except that instead of creating a static crypto map, you create a dynamic crypto map entry. You can also combine static and dynamic map entries within a single crypto map set.

Create a crypto dynamic map entry as follows:

Step 1 (Optional) Assign an access list to a dynamic crypto map:

crypto dynamic-map *dynamic-map-name dynamic-seq-num* **match address** *access-list-name*

This determines which traffic should be protected and not protected.

For example:

```
crypto dynamic-map dyn1 10 match address 101
```

In this example, access list 101 is assigned to dynamic crypto map "dyn1." The map sequence number is 10.

Step 2 Specify which transform sets are allowed for this dynamic crypto map. List multiple transform sets in order of priority (highest priority first).

crypto dynamic-map *dynamic-map-name dynamic-seq-num* **set transform-set** *transform-set-name1*, [*transform-set-name2*, …*transform-set-name9*]

For example:

```
crypto dynamic-map dyn 10 set transform-set myset1 myset2
```

In this example, when traffic matches access list 101, the SA can use either "myset1" (first priority) or "myset2" (second priority), depending on which transform set matches the transform sets of the peer.

Step 3 (Optional) Specify the SA lifetime for the crypto dynamic map entry if you want to override the global lifetime value:

crypto dynamic-map *dynamic-map-name dynamic-seq-num* **set security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes*}

For example:

```
crypto dynamic-map dyn1 10 set security-association lifetime seconds 2700
```

This example shortens the timed lifetime for dynamic crypto map "dyn1 10" to 2700 seconds (45 minutes). The time volume lifetime is not changed.

Step 4 (Optional) Specify that IPsec ask for PFS when requesting new SAs for this dynamic crypto map, or should demand PFS in requests received from the peer:

crypto dynamic-map *dynamic-map-name dynamic-seq-num* **set pfs** [**group1** | **group2** | **group5** | **group7**]

For example:

```
crypto dynamic-map dyn1 10 set pfs group5
```

Step 5 Add the dynamic crypto map set into a static crypto map set.

Be sure to set the crypto maps referencing dynamic maps to be the lowest priority entries (highest sequence numbers) in a crypto map set.

crypto map *map-name seq-num* **ipsec-isakmp dynamic** *dynamic-map-name*

For example:

```
crypto map mymap 200 ipsec-isakmp dynamic dyn1
```

Providing Site-to-Site Redundancy

You can define multiple peers by using crypto maps to provide redundancy. This configuration is useful for site-to-site VPNs.

If one peer fails, the security appliance establishes a tunnel to the next peer associated with the crypto map. It sends data to the peer that it has successfully negotiated with, and that peer becomes the "active" peer. The "active" peer is the peer that the security appliance keeps trying first for follow-on negotiations until a negotiation fails. At that point the security appliance goes on to the next peer. The security appliance cycles back to the first peer when all peers associated with the crypto map have failed.

Viewing an IPsec Configuration

Table 29-5 lists commands you can enter to view information about your IPsec configuration.

Table 29-5 Commands to View IPsec Configuration Information

Command	Purpose
show running-configuration crypto	Displays the entire crypto configuration, including IPsec, crypto maps, dynamic crypto maps, and ISAKMP.
show running-config crypto ipsec	Displays the complete IPsec configuration.
show running-config crypto isakmp	Displays the complete ISAKMP configuration.
show running-config crypto map	Displays the complete crypto map configuration.
show running-config crypto dynamic-map	Displays the dynamic crypto map configuration.
show all crypto map	View all of the configuration parameters, including those with default values.

Clearing Security Associations

Certain configuration changes take effect only during the negotiation of subsequent SAs. If you want the new settings to take effect immediately, clear the existing SAs to reestablish them with the changed configuration. If the security appliance is actively processing IPsec traffic, clear only the portion of the SA database that the configuration changes affect. Reserve clearing the full SA database for large-scale changes, or when the security appliance is processing a small amount of IPsec traffic.

Table 29-6 lists commands you can enter to clear and reinitialize IPsec SAs.

Table 29-6 Commands to Clear and Reinitialize IPsec SAs

Command	Purpose
clear configure crypto	Removes an entire crypto configuration, including IPsec, crypto maps, dynamic crypto maps, and ISAKMP.
clear configure crypto ca trustpoint	Removes all trustpoints.
clear configure crypto dynamic-map	Removes all dynamic crypto maps. Includes keywords that let you remove specific dynamic crypto maps.

Table 29-6 Commands to Clear and Reinitialize IPsec SAs (continued)

Command	Purpose
clear configure crypto map	Removes all crypto maps. Includes keywords that let you remove specific crypto maps.
clear configure crypto isakmp	Removes the entire ISAKMP configuration.
clear configure crypto isakmp policy	Removes all ISAKMP policies or a specific policy.
clear crypto isakmp sa	Removes the entire ISAKMP SA database.

Clearing Crypto Map Configurations

The **clear configure crypto** command includes arguments that let you remove elements of the crypto configuration, including IPsec, crypto maps, dynamic crypto maps, CA trustpoints, all certificates, certificate map configurations, and ISAKMP.

Be aware that if you enter the **clear configure crypto** command without arguments, you remove the entire crypto configuration, including all certificates.

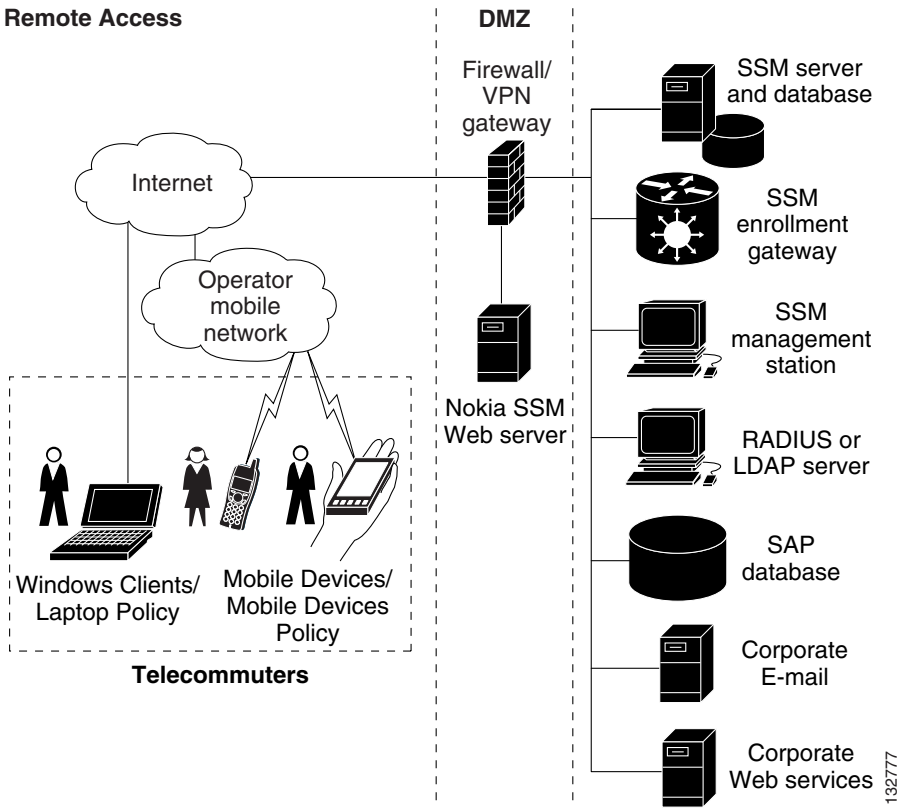
For more information, see the **clear configure crypto** command in the *Cisco Security Appliance Command Reference*.

Supporting the Nokia VPN Client

The security appliance supports connections from Nokia VPN Clients on Nokia 92xx Communicator series phones using the Challenge/Response for Authenticated Cryptographic Keys (CRACK) protocol. CRACK is ideal for mobile IPsec-enabled clients that use legacy authentication techniques instead of digital certificates. It provides mutual authentication when the client uses a legacy based secret-key authentication technique such as RADIUS and the gateway uses public-key authentication.

The Nokia back-end services must be in place to support both Nokia clients and the CRACK protocol. This requirement includes the Nokia Security Services Manager (NSSM) and Nokia databases as shown in Figure 29-5.

Figure 29-5 Nokia 92xx Communicator Service Requirement



To support the Nokia VPN Client, perform the following step on the security appliance:

- Enable CRACK authentication using the **crypto isakmp policy** *priority* **authentication** command with the **crack** keyword in global configuration mode. For example:

```
hostname(config)# crypto isakmp policy 2
hostname(config-isakmp-policy)# authentication crack
```

If you are using digital certificates for client authentication, perform the following additional steps:

Step 1 Configure the trustpoint and remove the requirement for a fully qualified domain name. The trustpoint might be NSSM or some other CA. In this example, the trustpoint is named CompanyVPNCA:

```
hostname(config)# crypto ca trustpoint CompanyVPNCA
hostname(config-ca-trustpoint)# fqdn none
```

Step 2 To configure the identity of the ISAKMP peer, perform one of the following steps:

a. Use the **crypto isakmp identity** command with the **hostname** keyword. For example:

```
hostname(config)# crypto isakmp identity hostname
```

–or–

b. Use the **crypto isakmp identity** command with the **auto** keyword to configure the identity to be automatically determined from the connection type. For example:

```
hostname(config)# crypto isakmp identity auto
```

Note If you use the **crypto isakmp identity auto** command, you must be sure that the DN attribute order in the client certificate is CN, OU, O, C, St, L.

To learn more about the Nokia services required to support the CRACK protocol on Nokia clients, and to ensure they are installed and configured properly, contact your local Nokia representative.



CHAPTER 30

Configuring L2TP over IPSec

This chapter describes how to configure IPSec over L2TP on the security appliance, and includes the following topics:

- L2TP Overview, page 30-1
- Configuring L2TP over IPSec Connections, page 30-3
- Viewing L2TP over IPSec Connection Information, page 30-6

L2TP Overview

Layer 2 Tunneling Protocol (L2TP) is a VPN tunneling protocol which allows remote clients to use the public IP network to securely communicate with private corporate network servers. L2TP uses PPP over UDP (port 1701) to tunnel the data.

L2TP protocol is based on the client/server model. The function is divided between the L2TP Network Server (LNS), and the L2TP Access Concentrator (LAC). The LNS typically runs on a network gateway such as a router, while the LAC can be a dial-up Network Access Server (NAS), or a PC with a bundled L2TP client such as Microsoft Windows 2000.

The primary benefit of configuring L2TP with IPSec in a remote access scenario is that remote users can access a VPN over a public IP network without a gateway or a dedicated line, enabling remote access from virtually anyplace with POTS. An additional benefit is that the only client requirement for VPN access is the use of Windows 2000 with Microsoft Dial-Up Networking (DUN). No additional client software, such as Cisco VPN client software, is required.

To configure L2TP over IPSec, first configure IPSec transport mode to enable IPSec with L2TP. Then configure L2TP with a virtual private dial-up network VPDN group.

The configuration of L2TP with IPSec supports certificates using the pre-shared keys or RSA signature methods, and the use of dynamic (as opposed to static) crypto maps. This summary of tasks assumes completion of IKE, as well as pre-shared keys or RSA signature configuration. See "Chapter 41, "Configuring Certificates,"" for the steps to configure pre-shared keys, RSA, and dynamic crypto maps.

Note L2TP with IPSec on the security appliance allows the LNS to interoperate with the Windows 2000 L2TP client. Interoperability with LACs from Cisco and other vendors is currently not supported. Only L2TP with IPSec is supported, native L2TP itself is not supported on security appliance.

Note The minimum IPSec security association lifetime supported by the Windows 2000 client is 300 seconds. If the lifetime on thesecurity appliance is set to less than 300 seconds, the Windows 2000 client ignores it and replaces it with a 300 second lifetime.

Note L2TP over IPsec sessions are not supported by stateful failover.

IPSec Transport and Tunnel Modes

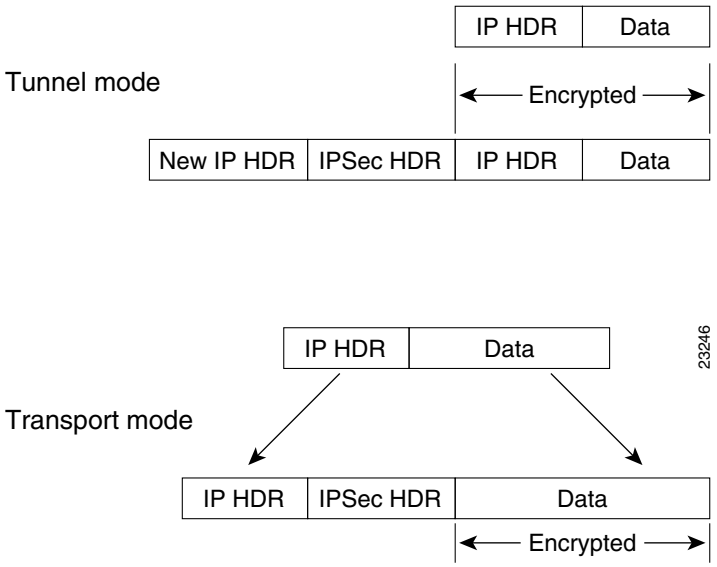
By default, the security appliance uses IPSec tunnel mode—the entire original IP datagram is encrypted, and it becomes the payload in a new IP packet. This mode allows a network device, such as a router, to act as an IPSec proxy. That is, the router performs encryption on behalf of the hosts. The source router encrypts packets and forwards them along the IPSec tunnel. The destination router decrypts the original IP datagram and forwards it on to the destination system. The major advantage of tunnel mode is that the end systems do not need to be modified to receive the benefits of IPSec. Tunnel mode also protects against traffic analysis; with tunnel mode, an attacker can only determine the tunnel endpoints and not the true source and destination of the tunneled packets, even if they are the same as the tunnel endpoints.

However, the Windows 2000 L2TP/IPSec client uses IPSec transport mode—only the IP payload is encrypted, and the original IP headers are left intact. This mode has the advantages of adding only a few bytes to each packet and allowing devices on the public network to see the final source and destination of the packet. Figure 30-1 illustrates the differences between IPSec Tunnel and Transport modes.

Therefore, In order for Windows 2000 L2TP/IPSec clients to connect to the security appliance, you must configure IPSec transport mode for a transform set using the **crypto ipsec transform-set trans_name mode transport** command. This command is the configuration procedure that follows, "Configuring L2TP over IPSec Connections" section on page 30-3**.**

With this capability (transport), you can enable special processing (for example, QoS) on the intermediate network based on the information in the IP header. However, the Layer 4 header will be encrypted, limiting the examination of the packet. Unfortunately, transmitting the IP header in clear text, transport mode allows an attacker to perform some traffic analysis.

Figure 30-1 IPSec in Tunnel and Transport Modes



Configuring L2TP over IPSec Connections

To configure the security appliance to accept L2TP over IPSec connections, follow these steps:

Note The security appliance does not establish an L2TP/IPSec tunnel with Windows 2000 if either the Cisco VPN Client Version 3.x or the Cisco VPN 3000 Client Version 2.5 is installed. Disable the *Cisco VPN Service* for the Cisco VPN Client Version 3.x, or the *ANetIKE Service* for the Cisco VPN 3000 Client Version 2.5 from the Services panel in Windows 2000 (click **Start>Programs>Administrative Tools>Services**). Then restart the IPSec Policy Agent Service from the **Services** panel, and reboot the machine.

Step 1 Specify IPSec to use transport mode rather than tunnel mode with the **mode** keyword of the **crypto ipsec transform-set** command:

```
hostname(config)# crypto ipsec transform-set trans_name mode transport
```

Step 2 (Optional) Specify the local address pool used to allocate the IP address to the client using the **address-pool** command in tunnel-group general-attributes mode:

```
hostname(config)# tunnel-group name general-attributes
hostname(config-tunnel-general)# address-pool pool_name
```

Step 3 (Optional) Instruct the security appliance to send DNS server IP addresses to the client with the **dns value** command from group policy configuration mode:

```
hostname(config)# group-policy group_policy_name attributes
hostname(config-group-policy)# dns value [none | IP_primary [IP_secondary]]
```

Step 4 (Optional) Instruct the security appliance to send WINS server IP addresses to the client using the **wins-server** command from group policy configuration mode:

```
hostname(config-group-policy)# wins-server value [none | IP_primary [IP_secondary]]
```

Step 5 (Optional) Generate a AAA accounting start and stop record for an L2TP session using the **accounting-server-group** command from tunnel group general-attributes mode:

```
hostname(config)# tunnel-group name general-attributes
hostname(config-tunnel-general)# accounting-server-group aaa_server_group
```

Step 6 Configure L2TP over IPSec as a valid VPN tunneling protocol for a group or user with the **vpn-tunnel-protocol l2tp-ipsec command**:

For a group, enter group-policy attributes mode:

```
hostname(config)# group-policy group_policy_name attributes
hostname(config-group-policy)# vpn-tunnel-protocol l2tp-ipsec
```

For a user, enter username attributes mode:

```
hostname(config)# username user_name attributes
hostname(config-username)# vpn-tunnel-protocol l2tp-ipsec
```

Step 7 Create a tunnel group with the **tunnel-group** command, and link the name of the group policy to the tunnel group with the **default-group-policy** command from tunnel group general-attributes mode:

```
hostname(config)# tunnel-group name type ipsec-ra
hostname(config)# tunnel-group name general-attributes
hostname(config-tunnel-general)# group-policy group_policy_name
```

Step 8 Configure the PPP authentication protocol using the **authentication** *type* command from tunnel group ppp-attributes mode. Table 30-1 shows the types of PPP authentication, and their characteristics.

```
hostname(config)# tunnel-group name ppp-attributes
hostname(config-ppp)# authentication pap
```

Table 30-1 Authentication Type Characteristics

Keyword	Authentication Type	Characteristics
chap	CHAP	In response to the server challenge, the client returns the encrypted [challenge plus password] with a cleartext username. This protocol is more secure than the PAP, but it does not encrypt data.
eap-proxy	EAP	Enables EAP which permits the security appliance to proxy the PPP authentication process to an external RADIUS authentication server.
ms-chap-v1 **ms-chap-v2**	Microsoft CHAP, Version 1 Microsoft CHAP, Version, 2	Similar to CHAP but more secure in that the server stores and compares only encrypted passwords rather than cleartext passwords as in CHAP. This protocol also generates a key for data encryption by MPPE.
pap	PAP	Passes cleartext username and password during authentication and is not secure.

Step 9 Specify a method to authenticate users attempting L2TP over IPSec connections. Use the **authentication-server-group** command from tunnel-group general-attributes mode to configure the security appliance to use an authentication server or its own local database.

Using an Authentication Server

To use an authentication server, use the **authentication server group** keyword:

```
hostname(config)# tunnel-group name general-attributes
hostname(config-tunnel-general)# authentication-server-group auth_server_group
```

Using the Local Database

To use the local database, enter the **LOCAL** keyword.

```
hostname(config)# tunnel-group name general-attributes
hostname(config-tunnel-general)# authentication-server-group LOCAL
```

Note The security appliance only supports the PPP authentications PAP and Microsoft CHAP, Versions 1 and 2, on the local database. EAP and CHAP are performed by proxy authentication servers. Therefore, if a remote user belongs to a tunnel group configured with the **authentication eap-proxy** or **authentication chap** commands, and the security appliance is configured to use the local database, that user will not be able to connect.

Step 10 Create a user in the local database with the **username** command from global configuration mode.

If the user is an L2TP client using Microsoft CHAP, Version 1 or Version 2, and the security appliance is configured to authenticate against the local database, you must include the **mschap** keyword. For Example:

```
hostname(config)# username t_wmith password eu5d93h mschap
```

Step 11 Configure the interval (in seconds) between hello messages using the **l2tp tunnel hello** command in global configuration mode:

```
hostname(config)# l2tp tunnel hello seconds
```

Step 12 (Optional) If you expect multiple L2TP clients behind a NAT device to attempt L2TP over IPSec connections to the security appliance, you must enable NAT traversal so that ESP packets can pass through one or more NAT devices.

To enable NAT traversal globally, check that ISAKMP is enabled (you can enable it with the **crypto isakmp enable** command) in global configuration mode and then use the **crypto isakmp nat-traversal** command. For example:

```
hostname(config)# crypto isakmp enable
hostname(config)# crypto isakmp nat-traversal 30
```

Tunnel Group Switching

Tunnel Group Switching enables the security appliance to associate different users that are establishing L2TP over IPSec connections with different tunnel groups. Since each tunnel group has its own AAA server group and IP address pools, users can be authenticated through methods specific to their tunnel group.

With this feature, instead of sending just a username, the user sends a username and a group name in the format *username@group_name, where "@" represents a delimiter that you can configure, and the group name is the name of a tunnel group that has been configured on the security appliance.*

To enable Tunnel Group Switching, you must enable Strip Group processing using the **strip-group** command from tunnel-group general-attributes mode. When enabled, the security appliance selects the tunnel group for user connections by obtaining the group name from the username presented by the VPN client. The security appliance then sends only the user part of the username for authorization and authentication. Otherwise (if disabled), the security appliance sends the entire username, including the realm. In the following example, Strip Group processing is enabled for the tunnel-group *telecommuters*:

```
asa1(config)# tunnel-group telecommuters general-attributes
asa1(config-tunnel-general)# strip-group
```

Apple iPhone and MAC OS X Compatibility

The security appliance requires the following IKE (ISAKMP) policy settings for successful Apple iPhone or MAC OS X connections:

- IKE phase 1—3DES encryption with SHA1 hash method.
- IPSec phase 2—3DES or AES encryption with MD5 or SHA hash method.
- PPP Authentication—PAP, MS-CHAPv1, or MSCHAPv2 (preferred).
- Pre-shared key (only for iPhone).

The following example shows configuration file commands that ensure iPhone and OS X compatibility:

```
tunnel-group DefaultRAGroup general-attributes
    address-pool pool
tunnel-group DefaultRAGroup ipsec-attributes
 pre-shared-key *
tunnel-group DefaultRAGroup ppp-attributes
    no authentication pap
    authentication chap
    authentication ms-chap-v1
    authentication ms-chap-v2
crypto ipsec transform-set trans esp-3des esp-sha-hmac
crypto ipsec transform-set trans mode transport
crypto dynamic-map dyno 10 set transform-set set trans
crypto map vpn 20 ipsec-isakmp dynamic dyno
crypto map vpn interface outside
crypto isakmp identity auto
crypto isakmp enable outside
crypto isakmp policy 10
    authentication pre-share
    encryption 3des
    hash sha
    group 2
    lifetime 86400
crypto isakmp nat-traversal 3600
```

For more information about setting IKE policies, see the *Configuring IPSec and ISAKMP.*

Viewing L2TP over IPSec Connection Information

The **show vpn-sessiondb** command includes protocol filters that you can use to view detailed information about L2TP over IPSec connections. The full command from global configuration mode is **show vpn-sessoindb detailed remote filter protocol l2tpOverIpsec.**

The following example shows the details of a single L2TP over IPSec connection:

```
hostname# show vpn-sessiondb detail remote filter protocol L2TPOverIPSec

Session Type: Remote Detailed

Username     : b_smith
Index        : 1
Assigned IP  : 90.208.1.200          Public IP    : 70.208.1.212
Protocol     : L2TPOverIPSec         Encryption   : 3DES
Hashing      : SHA1
Bytes Tx     : 418464                Bytes Rx     : 424440
Client Type  :                       Client Ver   :
Group Policy : DfltGrpPolicy
Tunnel Group : DefaultRAGroup
```

```
Login Time   : 13:24:48 UTC Thu Mar 30 2006
Duration     : 1h:09m:18s
Filter Name  : #ACSACL#-IP-ACL4Clients-440fa5aa
NAC Result   : N/A
Posture Token:

IKE Sessions: 1
IPSec Sessions: 1
L2TPOverIPSec Sessions: 1


IKE:
  Session ID   : 1
  UDP Src Port : 500                    UDP Dst Port : 500
  IKE Neg Mode : Main                   Auth Mode    : preSharedKeys
  Encryption   : 3DES                   Hashing      : SHA1
  Rekey Int (T): 28800 Seconds          Rekey Left(T): 24643 Seconds
  D/H Group    : 2

IPSec:
  Session ID   : 2
  Local Addr   : 80.208.1.2/255.255.255.255/17/1701
  Remote Addr  : 70.208.1.212/255.255.255.255/17/1701
  Encryption   : 3DES                   Hashing      : SHA1
  Encapsulation: Transport
  Rekey Int (T): 3600 Seconds           Rekey Left(T): 2856 Seconds
  Rekey Int (D): 95000 K-Bytes          Rekey Left(D): 95000 K-Bytes
  Idle Time Out: 30 Minutes             Idle TO Left : 30 Minutes
  Bytes Tx     : 419064                 Bytes Rx     : 425040
  Pkts Tx      : 4201                   Pkts Rx      : 4227


L2TPOverIPSec:
  Session ID   : 3
  Username     : l2tp
  Assigned IP  : 90.208.1.200
  Encryption   : none                   Auth Mode    : PAP
  Idle Time Out: 30 Minutes             Idle TO Left : 30 Minutes
  Bytes Tx     : 301386                 Bytes Rx     : 306480
  Pkts Tx      : 4198                   Pkts Rx      : 4224
```

The following example shows the details of a single L2TP over IPSec over NAT connection:

```
hostname# show vpn-sessiondb detail remote filter protocol L2TPOverIPSecOverNAtT

Session Type: Remote Detailed

Username     : v_gonzalez
Index        : 2
Assigned IP  : 90.208.1.202           Public IP    : 70.208.1.2
Protocol     : L2TPOverIPSecOverNatT  Encryption   : 3DES
Hashing      : MD5
Bytes Tx     : 1009                   Bytes Rx     : 2241
Client Type  :                        Client Ver   :
Group Policy : DfltGrpPolicy
Tunnel Group : l2tpcert
Login Time   : 14:35:15 UTC Thu Mar 30 2006
Duration     : 0h:00m:07s
Filter Name  :
NAC Result   : N/A
Posture Token:

IKE Sessions: 1
```

```
IPSecOverNatT Sessions: 1
L2TPOverIPSecOverNatT Sessions: 1


IKE:
  Session ID   : 1
  UDP Src Port : 4500                  UDP Dst Port : 4500
  IKE Neg Mode : Main                  Auth Mode    : rsaCertificate
  Encryption   : 3DES                  Hashing      : MD5
  Rekey Int (T): 300 Seconds           Rekey Left(T): 294 Seconds
  D/H Group    : 2

IPSecOverNatT:
  Session ID   : 2
  Local Addr   : 80.208.1.2/255.255.255.255/17/1701
  Remote Addr  : 70.208.1.2/255.255.255.255/17/0
  Encryption   : 3DES                  Hashing      : MD5
  Encapsulation: Transport
  Rekey Int (T): 300 Seconds           Rekey Left(T): 293 Seconds
  Idle Time Out: 1 Minutes             Idle TO Left : 1 Minutes
  Bytes Tx     : 1209                  Bytes Rx     : 2793
  Pkts Tx      : 20                    Pkts Rx      : 32


L2TPOverIPSecOverNatT:
  Session ID   : 3
  Username     : v_gonzalez
  Assigned IP  : 90.208.1.202
  Encryption   : none                  Auth Mode    : PAP
  Idle Time Out: 1 Minutes             Idle TO Left : 1 Minutes
  Bytes Tx     : 584                   Bytes Rx     : 2224
  Pkts Tx      : 18                    Pkts Rx      : 30
====================
```

Using L2TP Debug Commands

You can display L2TP debug information using the **debug l2tp** command in privileged EXEC mode. To disable the display of debug information, use the **no** form of this command:

debug l2tp {**data** | **error** | **event** | **packet**} *level*

data displays data packet trace information.

error displays error events.

event displays L2TP connection events.

packet displays packet trace information.

level sets the debug message level to display, between 1 and 255. The default is 1. To display additional messages at higher levels, set the level to a higher number.

The following example enables L2TP debug messages for connection events. The **show debug** command reveals that L2TP debug messages are enabled.

```
hostname# debug l2tp event 1
hostname# show debug
debug l2tp event enabled at level 1
hostname#
```

Enabling IPSec Debug

IPSec debug information can be added to a Windows 2000 client by adding the following registry:

Step 1 Run the Windows 2000 registry editor: REGEDIT.

Step 2 Locate the following registry entry:

MyComputer\HKEY_LOCAL_MACHINE\CurrentControlSet\Services\PolicyAgent

Step 3 Create the key by entering **`oakley`**.

Step 4 Create the DWORD by entering **`EnableLogging`**.

Step 5 Set the "Enable Logging" value to "1".

Step 6 Stop and Start the IPSec Policy Agent (click **Start>Programs>Administrative Tools>Services**). The debug file will be found at "%windir%\debug\oakley.log".

Getting Additional Information

Additional information on various topics can be found at www.microsoft.com:

http://support.microsoft.com/support/kb/articles/Q240/2/62.ASP

How to Configure an L2TP/IPSec Connection Using Pre-Shared Keys Authentication:

http://support.microsoft.com/support/kb/articles/Q253/4/98.ASP

How to Install a Certificate for Use with IP Security (IPSec):

http://www.microsoft.com/windows2000/en/server/help/default.asp?url=/WINDOWS2000/en/server/help/sag_VPN_us26.htm

How to use a Windows 2000 Machine Certificate for L2TP over IPSec VPN Connections:

http://www.microsoft.com/windows2000/techinfo/planning/security/ipsecsteps.asp#heading3

How to Create a Custom MMC Console and Enabling Audit Policy for Your Computer:

http://support.microsoft.com/support/kb/articles/Q259/3/35.ASP



CHAPTER 31

Setting General IPSec VPN Parameters

The security appliance implementation of virtual private networking includes useful features that do not fit neatly into categories. This chapter describes some of these features. It includes the following sections:

- Configuring VPNs in Single, Routed Mode, page 31-1
- Configuring IPSec to Bypass ACLs, page 31-1
- Permitting Intra-Interface Traffic, page 31-2
- Setting Maximum Active IPSec VPN Sessions, page 31-3
- Using Client Update to Ensure Acceptable Client Revision Levels, page 31-4
- Understanding Load Balancing, page 31-6
- Configuring Load Balancing, page 31-9
- Configuring VPN Session Limits, page 31-13

Configuring VPNs in Single, Routed Mode

VPNs work only in single, routed mode. VPN functionality is unavailable in configurations that include either security contexts, also referred to as multi-mode firewall, or Active/Active stateful failover.

The exception to this caveat is that you can configure and use one connection for administrative purposes to (not through) the security appliance in transparent mode.

Configuring IPSec to Bypass ACLs

To permit any packets that come from an IPSec tunnel without checking ACLs for the source and destination interfaces, enter the **sysopt connection permit-ipsec** command in global configuration mode.

You might want to bypass interface ACLs for IPSec traffic if you use a separate VPN concentrator behind the security appliance and want to maximize the security appliance performance. Typically, you create an ACL that permits IPSec packets using the **access-list** command and apply it to the source interface. Using an ACL is more secure because you can specify the exact traffic you want to allow through the security appliance.

The syntax is **sysopt connection permit-ipsec**. The command has no keywords or arguments.

The following example enables IPSec traffic through the security appliance without checking ACLs:

```
hostname(config)# sysopt connection permit-ipsec
```

Note Decrypted "through" traffic is permitted from the client despite having an access-group on the outside interface, which calls a "deny ip any any" access-list, while **no sysopt connection permit-vpn** is configured.

Users who want to control access to the protected network via Site-to-Site or remote access VPN using the **no sysopt permit** command in conjunction with an access control list (ACL) on the outside interface are not successful.

In this situation, when management-access inside is enabled, the ACL is not applied, and users can still connect using SSH to the security appliance. Traffic to hosts on the inside network are blocked correctly by the ACL, but can't block decrypted "through" traffic to the inside interface.

The **ssh** and **http** commands are of a higher priority than the ACLs. In other words, to deny ssh, telnet, or ICMP traffic to the box from the VPN session, use ssh, telnet and icmp commands, which denies the IP local pool should be added.

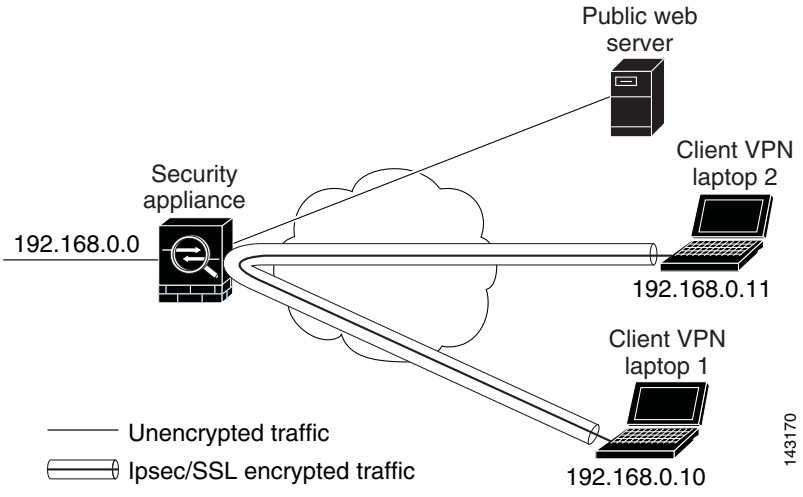
Permitting Intra-Interface Traffic

The security appliance includes a feature that lets a VPN client send IPSec-protected traffic to another VPN user by allowing such traffic in and out of the same interface. Also called “hairpinning”, this feature can be thought of as VPN spokes (clients) connecting through a VPN hub (security appliance).

In another application, this feature can redirect incoming VPN traffic back out through the same interface as unencrypted traffic. This would be useful, for example, to a VPN client that does not have split tunneling but needs to both access a VPN and browse the Web.

Figure 31-1 shows VPN Client 1 sending secure IPSec traffic to VPN Client 2 while also sending unencrypted traffic to a public Web server.

Figure 31-1 VPN Client Using Intra-Interface Feature for Hairpinning



To configure this feature, use the **same-security-traffic** command in global configuration mode with its **intra-interface** argument.

The command syntax is **same-security-traffic permit** {**inter-interface | intra-interface**}.

The following example shows how to enable intra-interface traffic:

```
hostname(config)# same-security-traffic permit intra-interface
hostname(config)#
```

Note You use the **same-security-traffic** command, but with the **inter-interface** argument, to permit communication between interfaces that have the same security level. This feature is not specific to IPSec connections. For more information, see the "Configuring Interface Parameters" chapter of this guide.

To use hairpinning, you must apply the proper NAT rules to the security appliance interface, as discussed in the following section.

NAT Considerations for Intra-Interface Traffic

For the security appliance to send unencrypted traffic back out through the interface, you must enable NAT for the interface so that publicly routable addresses replace your private IP addresses (unless you already use public IP addresses in your local IP address pool). The following example applies an interface PAT rule to traffic sourced from the client IP pool:

```
hostname(config)# ip local pool clientpool 192.168.0.10-192.168.0.100
hostname(config)# global (outside) 1 interface
hostname(config)# nat (outside) 1 192.168.0.0 255.255.255.0
```

When the security appliance sends encrypted VPN traffic back out this same interface, however, NAT is optional. The VPN-to-VPN hairpinning works with or without NAT. To apply NAT to all outgoing traffic, implement only the commands above. To exempt the VPN-to-VPN traffic from NAT, add commands (to the example above) that implement NAT exemption for VPN-to-VPN traffic, such as:

```
hostname(config)# access-list nonat permit ip 192.168.0.0 255.255.255.0 192.168.0.0
255.255.255.0
hostname(config)# nat (outside) 0 access-list nonat
```

For more information on NAT rules, see the "Applying NAT" chapter of this guide.

Setting Maximum Active IPSec VPN Sessions

To limit VPN sessions to a lower value than the security appliance allows, enter the **vpn-sessiondb max-session-limit** command in global configuration mode.

- This command applies to all types of VPN sessions, including WebVPN.
- This limit affects the calculated load percentage for VPN Load Balancing.

The syntax is **vpn-sessiondb max-session-limit {***session-limit***}.**

The following example shows how to set a maximum VPN session limit of 450:

```
hostname (config)# vpn-sessiondb max-session-limit 450
hostname (config)#
```

Using Client Update to Ensure Acceptable Client Revision Levels

The client update feature lets administrators at a central location automatically notify VPN client users that it is time to update the VPN client software and the VPN 3002 hardware client image.

Remote users might be using outdated VPN software or hardware client versions. You can use the **client-update** command at any time to enable updating client revisions; specify the types and revision numbers of clients to which the update applies; provide a URL or IP address from which to get the update; and, in the case of Windows clients, optionally notify users that they should update their VPN client version. For Windows clients, you can provide a mechanism for users to accomplish that update. For VPN 3002 hardware client users, the update occurs automatically, with no notification. This command applies only to the IPSec remote-access tunnel-group type.

To perform client update, enter the **client-update** command in either general configuration mode or tunnel-group ipsec-attributes configuration mode. If the client is already running a software version on the list of revision numbers, it does not need to update its software. If the client is not running a software version on the list, it should update. The following procedure tells how to perform a client-update:

Step 1 In global configuration mode, enable client update by entering the command:

```
hostname(config)# client-update enable
hostname(config)#
```

Step 2 In global configuration mode, specify the parameters for the client update that you want to apply to all clients of a particular type. That is, specify the type of client, the URL or IP address from which to get the updated image, and the acceptable revision number or numbers for that client. You can specify up to four revision numbers, separated by commas.

If the user's client revision number matches one of the specified revision numbers, there is no need to update the client. This command specifies the client-update values for all clients of the specified type across the entire security appliance

The syntax of the command to do this is:

```
hostname(config)# client-update type type url url-string rev-nums rev-numbers
hostname(config)#
```

The available client types are **win9X** (includes Windows 95, Windows 98 and Windows ME platforms), **winnt** (includes Windows NT 4.0, Windows 2000 and Windows XP platforms), **windows** (Includes all Windows based platforms), and **vpn3002** (VPN 3002 hardware client).

If the client is already running a software version on the list of revision numbers, it does not need to update its software. If the client is not running a software version on the list, it should update. You can specify up to three of these client update entries. The keyword **windows** covers all of the allowable Windows platforms. If you specify **windows**, do not specify the individual Windows client types.

Note For all Windows clients, you must use the protocol http:// or https:// as the prefix for the URL. For the VPN 3002 hardware client, you must specify protocol tftp:// instead.

The following example configures client update parameters for the remote-access tunnel-group. It designates the revision number, 4.6.1 and the URL for retrieving the update, which is https://support/updates:

```
hostname(config)# client-update type windows url https://support/updates/ rev-nums 4.6.1
hostname(config)#
```

Alternatively, you can configure client update just for individual tunnel-groups, rather than for all clients of a particular type. (See Step 3.)

VPN 3002 clients update without user intervention and users receive no notification message. The following example applies only to VPN 3002 Hardware Clients. Entered in tunnel-group ipsec-attributes configuration mode, it configures client update parameters for the IPSec remote-access tunnel-group "salesgrp". It designates the revision number, 4.7 and uses the TFTP protocol for retrieving the updated software from the site with the IP address 192.168.1.1:

```
hostname(config)# tunnel-group salesgrp type ipsec-ra
hostname(config)# tunnel-group salesgrp ipsec-attributes
hostname(config-tunnel-ipsec)# client-update type vpn3002 url tftp:192.168.1.1 rev-nums
4.7
hostname(config-tunnel-ipsec)#
```

Note You can have the browser automatically start an application by including the application name at the end of the URL; for example: **https://support/updates/vpnclient.exe.**

Step 3 To define a set of client-update parameters for a particular ipsec-ra tunnel group, do the following. In tunnel-group ipsec-attributes mode, specify the tunnel-group name and its type, the URL or IP address from which to get the updated image, and a revision number. If the user's client's revision number matches one of the specified revision numbers, there is no need to update the client; for example, for a Windows client:

```
hostname(config)# tunnel-group remotegrp type ipsec-ra
hostname(config)# tunnel-group remotegrp ipsec-attributes
hostname(config-tunnel-ipsec)# client-update type windows url https://support/updates/
rev-nums 4.6.1
hostname(config-tunnel-ipsec)#
```

Step 4 Optionally, you can send a notice to active users with outdated Windows clients that their client needs updating. For these users, a pop-up window appears, offering them the opportunity to launch a browser and download the updated software from the site that you specified in the URL. The only part of this message that you can configure is the URL. (See Step 2 or 3.) Users who are not active get a notification message the next time they log on. You can send this notice to all active clients on all tunnel groups, or you can send it to clients on a particular tunnel group. For example, to notify all active clients on all tunnel groups, you would enter the following command in privileged EXEC mode:

```
hostname# client-update all
hostname#
```

If the user's client's revision number matches one of the specified revision numbers, there is no need to update the client, and no notification message is sent to the user. VPN 3002 clients update without user intervention and users receive no notification message.

Note If you specify the client-update type as **windows** (specifying all Windows-based platforms) and later want to enter a client-update type of **win9x** or **winnt** for the same entity, you must first remove the windows client type with the **no** form of the command, then use new client-update commands to specify the new client types.

Understanding Load Balancing

If you have a remote-access configuration in which you are using two or more security appliances or VPN Concentrators connected on the same network to handle remote sessions, you can configure these devices to share their session load. This feature is called *load balancing*. To implement load balancing, you group together logically two or more devices on the same private LAN-to-LAN network, private subnet, and public subnet into a *virtual cluster.*

All devices in the virtual cluster carry session loads. Load balancing directs session traffic to the least loaded device in the cluster, thus distributing the load among all devices. It makes efficient use of system resources and provides increased performance and high availability.

One device in the virtual cluster, the *virtual cluster master*, directs incoming traffic to the other devices, called *secondary devices*. The virtual cluster master monitors all devices in the cluster, keeps track of how busy each is, and distributes the session load accordingly. The role of virtual cluster master is not tied to a physical device; it can shift among devices. For example, if the current virtual cluster master fails, one of the secondary devices in the cluster takes over that role and immediately becomes the new virtual cluster master.

Note The output of a **show** command might show the secondary devices in the cluster as backup devices.

The virtual cluster appears to outside clients as a single *virtual cluster IP address*. This IP address is not tied to a specific physical device. It belongs to the current virtual cluster master; hence, it is virtual. A VPN Client attempting to establish a connection connects first to this virtual cluster IP address. The virtual cluster master then sends back to the client the public IP address of the least-loaded available host in the cluster. In a second transaction (transparent to the user), the client connects directly to that host. In this way, the virtual cluster master directs traffic evenly and efficiently across resources.

Note All clients other than the Cisco VPN Client or the Cisco 3002 Hardware Client should connect directly to the security appliance as usual; they do not use the virtual cluster IP address.

If a machine in the cluster fails, the terminated sessions can immediately reconnect to the virtual cluster IP address. The virtual cluster master then directs these connections to another active device in the cluster. Should the virtual cluster master itself fail, a secondary device in the cluster immediately and automatically takes over as the new virtual session master. Even if several devices in the cluster fail, users can continue to connect to the cluster as long as any one device in the cluster is up and available.

Implementing Load Balancing

Enabling load balancing involves:

- Configuring the load-balancing cluster by establishing a common virtual cluster IP address, UDP port (if necessary), and IPSec shared secret for the cluster. These values are should be configured indentically for every device in the cluster.
- Configuring the participating device by enabling load balancing on the device and defining device-specific properties. These values vary from device to device.

Note VPN load balancing requires an active 3DES/AES license. The security appliance checks for the existence of this crypto license before enabling load balancing. If it does not detect an active 3DES or AES license, the security appliance prevents the enabling of load balancing and also prevents internal configuration of 3DES by the load balancing system unless the license permits this usage.

Prerequisites

Load balancing is disabled by default. You must explicitly enable load balancing.

You must have first configured the public (outside) and private (inside) interfaces and also have previously configured the the interface to which the virtual cluster IP address refers. You can use the **interface** and **nameif** commands to configure different names for these interfaces. Subsequent references in this section use the names outside and inside.

All devices that participate in a cluster must share the same cluster-specific values: IP address, encryption settings, encryption key, and port.

Eligible Platforms

A load-balancing cluster can include security appliance models ASA 5510 (with a Plus license) and Model 5520 and above. You can also include VPN 3000 Series Concentrators in the cluster. While mixed configurations are possible, administration is generally simpler if the cluster is homogeneous.

Eligible Clients

Load balancing is effective only on remote sessions initiated with the following clients:

- Cisco AnyConnect VPN Client (Release 2.0 and later)
- Cisco VPN Client (Release 3.0 and later)
- Cisco VPN 3002 Hardware Client (Release 3.5 or later)
- Cisco PIX 501/506E when acting as an Easy VPN client.

Load balancing works with both IPSec clients and WebVPN sessions. All other clients, including LAN-to-LAN connections, can connect to a security appliance on which load balancing is enabled, but they cannot participate in load balancing.

VPN Load-Balancing Cluster Configurations

A load-balancing cluster can consist of all ASA Release 7.0(x) security appliances, all ASA Release 7.1(1) security appliances, all VPN 3000 Concentrators, or a mixture of these, subject to the following restrictions:

- Load-balancing clusters that consist of all ASA 7.0(x) security appliances, all ASA 7.1(1) security appliances, or all VPN 3000 Concentrators can run load balancing for a mixture of IPSec and WebVPN sessions.
- Load-balancing clusters that consist of a both of ASA 7.0(x) security appliances and VPN 3000 Concentrators can run load balancing for a mixture of IPSec and WebVPN sessions.

- Load-balancing clusters that include ASA 7.1(1) security appliances and either ASA 7.0(x) or VPN 3000 Concentrators or both can support only IPSec sessions. In such a configuration, however, the ASA 7.1(1) security appliances might not reach their full IPSec capacity. "Scenario 1: Mixed Cluster with No WebVPN Connections" on page 8, illustrates this situation.

With Release 7.1(1), IPSec and WebVPN sessions count or weigh equally in determining the load that each device in the cluster carries. This represents a departure from the load balancing calculation for the ASA Release 7.0(x) software and the VPN 3000 Concentrator, in that these platforms both use a weighting algorithm that, on some hardware platforms, calculates WebVPN session load differently from IPSec session load.

The virtual master of the cluster assigns session requests to the members of the cluster. An ASA Release 7.1(1) security appliance regards all sessions, WebVPN or IPSec, as equal and assigns them accordingly. An ASA Release 7.0(x) security appliance or a VPN 3000 Concentrator performs a weighting calculation in assigning session loads.

Note You can configure the number of IPSec and WebVPN sessions to allow, up to the maximum allowed by your configuration and license. See the "Configuring VPN Session Limits" section on page 31-13 for a description of how to set these limits.

Some Typical Mixed Cluster Scenarios

If you have a mixed configuration—that is, if your load-balancing cluster includes devices running a mixture of ASA software releases or at least one security appliance running ASA Release 7.1(1) and a VPN 3000 Concentrator—the difference in weighting algorithms becomes an issue if the initial cluster master fails and another device takes over as master.

The following scenarios illustrate the use of VPN load balancing in clusters consisting of a mixture of security appliances running ASA Release 7.1(1) and ASA Release 7.0(x) software, as well as VPN 3000 Series Concentrators.

Scenario 1: Mixed Cluster with No WebVPN Connections

In this scenario, the cluster consists of a mixture of security appliances and VPN 3000 Concentrators. Some of the security appliance cluster peers are running ASA Release 7.0(x), and some are running Release 7.1(1). The pre-7.1(1) and VPN 3000 peers do not have any SSL VPN connections, and the 7.1(1) cluster peers have only the base SSL VPN license, which allows two WebVPN sessions, but there are no SSL VPN connections. In this case, all the connections are IPSec, and load balancing works fine.

The two WebVPN licenses have a very small effect on the user's taking advantage of the maximum IPSec session limit, and then only when a VPN 3000 Concentrator is the cluster master. In general, the smaller the number of WebVPN licenses is on a security appliance in a mixed cluster, the smaller the effect on the ASA 7.1(1) device being able to reach its IPSec session limit in a scenario where there are only IPSec sessions.

Scenario 2: Mixed Cluster Handling WebVPN Connections

Suppose, for example, a security appliance running ASA Release 7.1(1) software is the initial cluster master; then that device fails. Another device in the cluster takes over automatically as master and applies its own load-balancing algorithm to determine processor loads within the cluster. A cluster master running ASA Release 7.1(1) software cannot weight session loads in any way other than what that software provides. Therefore, it cannot assign a combination of IPSec and WebVPN session loads

properly to ASA devices running earlier versions nor to VPN 3000 Concentrators. Conversely, a VPN 3000 Concentrator acting as the cluster master cannot assign loads properly to an ASA Release 7.1(1) security appliance. The following scenario illustrates this dilemma.

This scenario is similar to the previous one, in that the cluster consists of a mixture of security appliances and VPN 3000 Concentrators. Some of the security appliance cluster peers are running ASA Release 7.0,(x) and some are running Release 7.1(1). In this case, however, the cluster is handling SSL VPN connections as well as IPSec connections.

If a device that is running software earlier than ASA Release 7.1(1) is the cluster master, the master applies the protocol and logic in effect prior to Release 7.1(1). That is, sessions might be directed to load-balancing peers that have exceeded their session limit. In that case, the user is denied access.

If the cluster master is a device running ASA Release 7.0(x) software, the old session-weighting algorithm applies only to the pre-7.1(1) peers in the cluster. No one should be denied access in this case. Because the pre-7.1(1) peers use the session-weighting algorithm, they are more lightly loaded.

An issue arises, however, because you cannot guarantee that the 7.1(1) peer is always the cluster master. If the cluster master fails, another peer assumes the role of master. The new master might be any of the eligible peers. Because of the innately unpredictability of the results, we recommend that you avoid configuring this type of cluster.

Configuring Load Balancing

To use load balancing, configure the following elements for each device that participates in the cluster.

- Public and private interfaces
- VPN load-balancing cluster attributes

Note All participants in the cluster must have an identical cluster configuration, except for the device priority within the cluster.

Configuring the Public and Private Interfaces for Load Balancing

To configure the public (outside) and private (inside) interfaces for the load-balancing cluster devices, do the following steps:

Step 1 Configure the public interface on the security appliance by entering the **interface** command with the **lbpublic** keyword in vpn-load-balancing configuration mode. This command specifies the name or IP address of the public interface for load balancing for this device:

```
hostname(config)# vpn load-balancing
hostname(config-load-balancing)# interface lbpublic outside
hostname(config-load-balancing)#
```

Step 2 Configure the private interface on the security appliance by entering the **interface** command with the **lbprivate** keyword in vpn-load-balancing configuration mode. This command specifies the name or IP address of the private interface for load balancing for this device:

```
hostname(config-load-balancing)# interface lbprivate inside
hostname(config-load-balancing)#
```

Step 3 Set the priority to assign to this device within the cluster. The range is from 1 to 10. The priority indicates the likelihood of this device becoming the virtual cluster master, either at start-up or when an existing master fails. The higher you set the priority (for example, 10), the more likely it is that this device becomes the virtual cluster master.

```
hostname(config-load-balancing)# priority number
hostname(config-load-balancing)#
```

For example, to assign this device a priority of 6 within the cluster, enter the following command:

```
hostname(config-load-balancing)# priority 6
hostname(config-load-balancing)#
```

Step 4 If you want to apply network address translation for this device, enter the **nat** command with the NAT assigned address for the device:

```
hostname(config-load-balancing)# nat ip_address
hostname(config-load-balancing)#
```

For example, to assign this device a NAT address of 192.168.30.3, enter the following command:

```
hostname(config-load-balancing)# nat 192.168.30.3
hostname(config-load-balancing)#
```

Configuring the Load Balancing Cluster Attributes

To configure the load-balancing cluster attributes for each device in the cluster, do the following steps:

Step 1 Set up VPN load balancing by entering the vpn load-balancing command in global configuration mode:

```
hostname(config)# vpn load-balancing
hostname(config-load-balancing)#
```

This enters vpn-load-balancing configuration mode, in which you can configure the remaining load-balancing attributes.

Step 2 Configure the IP address of the cluster to which this device belongs. This command specifies the single IP address that represents the entire virtual cluster. Choose an IP address that is within the public subnet address range shared by all the security appliances in the virtual cluster

```
hostname(config-load-balancing)# cluster ip address ip_address
hostname(config-load-balancing)#
```

For example, to set the cluster IP address to 192.168.10.10, enter the following command:

```
hostname(config-load-balancing)# cluster ip address 192.168.10.10
hostname(config-load-balancing)#
```

Step 3 Configure the cluster port.This command specifies the UDP port for the virtual cluster in which this device is participating. The default value is 9023. If another application is using this port, enter the UDP destination port number you want to use for load balancing.

```
hostname(config-load-balancing)# cluster port port_number
hostname(config-load-balancing)#
```

For example, to set the cluster port to 4444, enter the following command:

```
hostname(config-load-balancing)# cluster port 4444
hostname(config-load-balancing)#
```

Step 4 Optionally, enable IPSec encryption for the cluster. The default is no encryption. This command enables or disables IPSec encryption. If you configure this check attribute, you must first specify and verify a shared secret.The security appliances in the virtual cluster communicate via LAN-to-LAN tunnels using IPSec. To ensure that all load-balancing information communicated between the devices is encrypted, enable this attribute.

```
hostname(config-load-balancing)# cluster encryption
hostname(config-load-balancing)#
```



Note When using encryption, you must have previously configured the load-balancing inside interface. If that interface is not enabled on the load-balancing inside interface, you get an error message when you try to configure cluster encryption.

If the load-balancing inside interface was enabled when you configured cluster encryption, but was disabled before you configured the participation of the device in the virtual cluster, you get an error message when you enter the **participate** command (or, in ASDM, select the Participate in Load Balancing Cluster check box), and encryption is not enabled for the cluster.

To use cluster encryption, you musts enable isakmp on the inside interface, using the **crypto isakmp enable** command with the inside interface specified.

Step 5 If you enable cluster encryption, you must also specify the IPSec shared secret by entering the **cluster key** command. This command specifies the shared secret to between IPSec peers when you have enabled IPSec encryption. The value you enter in the box appears as consecutive asterisk characters

```
hostname(config-load-balancing)# cluster key shared_secret
hostname(config-load-balancing)#
```

For example, to set the shared secret to 123456789, enter the following command:

```
hostname(config-load-balancing)# cluster key 123456789
hostname(config-load-balancing)#
```

Step 6 Enable this device's participation in the cluster by entering the participate command:

```
hostname(config-load-balancing)# participate
hostname(config-load-balancing)#
```

Enabling Redirection Using a Fully-qualified Domain Name

To enable or disable redirection using a fully-qualified domain name in vpn load-balancing mode, use the **redirect-fqdn enable** command in global configuration mode. This behavior is disabled by default.

By default, the ASA sends only IP addresses in load-balancing redirection to a client. If certificates are in use that are based on DNS names, the certificates will be invalid when redirected to a secondary device.

As a VPN cluster master, this security appliance can send a fully qualified domain name (FQDN), using reverse DNS lookup, of a cluster device (another security appliance in the cluster), instead of its outside IP address, when redirecting VPN client connections to that cluster device.

All of the outside and inside network interfaces on the load-balancing devices in a cluster must be on the same IP network.

To do WebVPN load Balancing using FQDNs rather than IP addresses, you must do the following configuration steps:

Step 1 Enable the use of FQDNs for Load Balancing with the **redirect-fqdn enable** command:

redirect-fqdn {enable | disable}
no redirect-fqdn {enable | disable}

For example,
```
hostname(config)# vpn load-balancing
hostname(config-load-balancing)# redirect-fqdn enable
hostname(config-load-balancing)#
```

Step 2 Add an entry for each of your ASA outside interfaces into your DNS server, if such entries are not already present. Each ASA outside IP address should have a DNS entry associated with it for lookups. These DNS entries must also be enabled for Reverse Lookup.

Step 3 Enable DNS lookups on your ASA with the command - "dns domain-lookup inside" (or whichever interface has a route to your DNS server).

Step 4 Define your DNS server IP address on the ASA; for example: `dns name-server 10.2.3.4` (IP address of your DNS server).

The following is an example of a VPN load-balancing command sequence that includes an interface command that enables redirection for a fully-qualified domain name, specifies the public interface of the cluster as "test" and the private interface of the cluster as "foo":

```
hostname(config)# interface GigabitEthernet 0/1
hostname(config-if)# ip address 209.165.202.159 255.255.255.0
hostname(config)# nameif test
hostname(config)# interface GigabitEthernet 0/2
hostname(config-if)# ip address 209.165.201.30 255.255.255.0
hostname(config)# nameif foo
hostname(config)# vpn load-balancing
hostname(config-load-balancing)# nat 192.168.10.10
hostname(config-load-balancing)# priority 9
hostname(config-load-balancing)# interface lbpublic test
hostname(config-load-balancing)# interface lbprivate foo
hostname(config-load-balancing)# cluster ip address 209.165.202.224
hostname(config-load-balancing)# cluster key 123456789
hostname(config-load-balancing)# cluster encryption
hostname(config-load-balancing)# cluster port 9023
hostname(config-load-balancing)# redirect-fqdn enable
hostname(config-load-balancing)# participate
```

Viewing Load Balancing

The load balancing cluster master receives a periodic message from each ASA in the cluster with the number of active AnyConnect and clientless sessions, as well as the maximum allowed sessions based on the configured or license limits. If an ASA in the cluster shows 100% full capacity, the cluster master cannot redirect more connections to it. Although the ASA may show as full, some users may be in Inactive/wait-to-resume state, wasting the licenses. As a workaround, each ASA provides the total number of sessions minus the sessions in inactive state, instead of the total number of sessions. (Refer to **show vpn-sessiondb summary** in the *Cisco Security Appliance Command Reference Guide*. In other words, the inactive sessions are not reported to the cluster master. Even if the ASA is full (with some

inactive sessions), the cluster master still redirects connections to it if necessary. When the ASA receives the new connection, the session that has been inactive the longest is logged off, allowing new connections to take its license.

The following example shows 100 SSL sessions (Active only) and a 2% SSL load. These numbers do not include the inactive sessions. In other words, inactive sessions do not count towards the load for load balancing.

```
nmeka-asa2# sh vpn load-balancing
    Status :       enabled
    Role :         Master
    Failover :     Active
    Encryption :   enabled
    Cluster IP :   192.168.1.100
    Peers :        1

                                              Load %
Sessions
    Public IP      Role    Pri     Model          IPsec   SSL     IPSec SSL
    192.168.1.9    Master  7       ASA-5540       4       2       216     100
    192.168.1.19   Backup  9       ASA-5520       0       0       0       0
```

Configuring VPN Session Limits

You can run as many IPSec and WebVPN sessions as your platform and license for the security appliance supports. To view the licensing information for your security appliance, enter the **show version** command in global configuration mode. The following example shows the command and the licensing information excerpted from the output of this command:

```
hostname(config)# show version


Cisco Adaptive Security Appliance Software Version 7.1(0)182
Device Manager Version 5.1(0)128

Licensed features for this platform:
Maximum Physical Interfaces : Unlimited
Maximum VLANs               : 100
Inside Hosts                : Unlimited
Failover                    : Active/Active
VPN-DES                     : Enabled
VPN-3DES-AES                : Enabled
Security Contexts           : 10
GTP/GPRS                    : Enabled
VPN Peers                   : 750
WebVPN Peers                : 500

This platform has an ASA 5520 VPN Plus license.
```

To limit the maximum number of active IPSec VPN sessions to a lower value than the security appliance allows, enter the **vpn-sessiondb max-session-limit** command in global configuration mode. This limit affects the calculated load percentage for VPN Load Balancing.

```
hostname(config)# vpn-sessiondb max-session-limit number_of_sessions
hostname(config)#
```

For example, if the security appliance license allows 750 IPSec sessions, and you want to limit the number of IPSec sessions to 500, enter the following command:

```
hostname(config)# vpn-sessiondb max-session-limit 500
```

```
hostname(config)#
```

To remove the session limit, use the **no** version of this command.:

```
hostname(config)# no vpn-sessiondb max-session-limit
hostname(config)#
```

To limit WebVPN sessions to a lower value than the security appliance allows, use the **vpn-sessiondb max-webvpn-session-limit** command in global configuration mode. To remove the session limit, use the **no** version of this command.

```
hostname(config)# vpn-sessiondb max-webvpn-session-limit number_of_sessions
hostname(config)#
```

For example, if the security appliance license allows 500 WebVPN sessions, and you want to limit the number of WebVPN sessions to 250, enter the following command:

```
hostname(config)# vpn-sessiondb max-webvpn-session-limit 250
hostname(config)#
```

To remove the session limit, use the **no** version of this command.:

```
hostname(config)# no vpn-sessiondb max-webvpn-session-limit
hostname(config)#
```

For a complete description of the features available with each license, see Appendix A, Feature Licenses and Specifications.



CHAPTER 32

Configuring Connection Profiles, Group Policies, and Users

This chapter describes how to configure VPN connection profiles (formerly called "tunnel groups"), group policies, and users. This chapter includes the following sections.

- Overview of Connection Profiles, Group Policies, and Users, page 32-1
- Configuring Connection Profiles, page 32-6
- Group Policies, page 32-36
- Configuring User Attributes, page 32-75

In summary, you first configure connection profiles to set the values for the connection. Then you configure group policies. These set values for users in the aggregate. Then you configure users, which can inherit values from groups and configure certain values on an individual user basis. This chapter describes how and why to configure these entities.

Overview of Connection Profiles, Group Policies, and Users

Groups and users are core concepts in managing the security of virtual private networks (VPNs) and in configuring the security appliance. They specify attributes that determine user access to and use of the VPN. A *group* is a collection of users treated as a single entity. *Users* get their attributes from *group policies*. *Connection profiles* identify the group policy for a specific connection. If you do not assign a particular group policy to a user, the default group policy for the connection applies.

Note You configure connection profiles using **tunnel-group** commands. In this chapter, the terms "connection profile" and "tunnel group" are often used interchangeably.

Connection profiles and group policies simplify system management. To streamline the configuration task, the security appliance provides a default LAN-to-LAN connection profile, a default remote access connection profile, a default connection profile for clientless SSL VPN, and a default group policy (DfltGrpPolicy). The default connection profiles and group policy provide settings that are likely to be common for many users. As you add users, you can specify that they "inherit" parameters from a group policy. Thus you can quickly configure VPN access for large numbers of users.

If you decide to grant identical rights to all VPN users, then you do not need to configure specific connection profiles or group policies, but VPNs seldom work that way. For example, you might allow a finance group to access one part of a private network, a customer support group to access another part,

and an MIS group to access other parts. In addition, you might allow specific users within MIS to access systems that other MIS users cannot access. Connection profiles and group policies provide the flexibility to do so securely.

Note The security appliance also includes the concept of object groups, which are a superset of network lists. Object groups let you define VPN access to ports as well as networks. Object groups relate to ACLs rather than to group policies and connection profiles. For more information about using object groups, see Chapter 18, "Identifying Traffic with Access Lists."

The security appliance can apply attribute values from a variety of sources. It applies them according to the following hierarchy:

1. Dynamic Access Policy (DAP) record
2. Username
3. Group policy
4. Group policy for the connection profile
5. Default group policy

Therefore, DAP values for an attribute have a higher priority than those configured for a user, group policy, or connection profile.

When you enable or disable an attribute for a DAP record, the security appliance applies that value and enforces it. For example, when you disable HTTP proxy in dap webvpn mode, the security appliance looks no further for a value. When you instead use the **no** value for the **http-proxy** command, the attribute is not present in the DAP record, so the security appliance moves down to the AAA attribute in the username, and if necessary, the group policy to find a value to apply. We recommend that you use ASDM to configure DAP.

Connection Profiles

A connection profile consists of a set of records that determines tunnel connection policies. These records identify the servers to which the tunnel user is authenticated, as well as the accounting servers, if any, to which connection information is sent. They also identify a default group policy for the connection, and they contain protocol-specific connection parameters. Connection profiles include a small number of attributes that pertain to creating the tunnel itself. Connection profiles include a pointer to a group policy that defines user-oriented attributes.

The security appliance provides the following default connection profiles: DefaultL2Lgroup for LAN-to-LAN connections, DefaultRAgroup for remote access connections, and DefaultWEBVPNGroup for clientless SSL VPN (browser-based) connections. You can modify these default connection profiles, but you cannot delete them. You can also create one or more connection profiles specific to your environment. Connection profiles are local to the security appliance and are not configurable on external servers.

Connection profiles specify the following attributes:

- General Connection Profile Connection Parameters, page 32-3
- IPSec Tunnel-Group Connection Parameters, page 32-4
- Connection Profile Connection Parameters for Clientless SSL VPN Sessions, page 32-5

General Connection Profile Connection Parameters

General parameters are common to all VPN connections. The general parameters include the following:

- Connection profile name—You specify a connection-profile name when you add or edit a connection profile. The following considerations apply:
  - For clients that use preshared keys to authenticate, the connection profile name is the same as the group name that an IPSec client passes to the security appliance.
  - Clients that use certificates to authenticate pass this name as part of the certificate, and the security appliance extracts the name from the certificate.
- Connection type—Connection types include IPSec remote access, IPSec LAN-to-LAN, and clientless SSL VPN. A connection profile can have only one connection type.
- Authentication, Authorization, and Accounting servers—These parameters identify the server groups or lists that the security appliance uses for the following purposes:
  - Authenticating users
  - Obtaining information about services users are authorized to access
  - Storing accounting records

  A server group can consist of one or more servers.
- Default group policy for the connection—A group policy is a set of user-oriented attributes. The default group policy is the group policy whose attributes the security appliance uses as defaults when authenticating or authorizing a tunnel user.
- Client address assignment method—This method includes values for one or more DHCP servers or address pools that the security appliance assigns to clients.
- Override account disabled—This parameter lets you override the “account-disabled” indicator received from a AAA server.
- Password management—This parameter lets you warn a user that the current password is due to expire in a specified number of days (the default is 14 days), then offer the user the opportunity to change the password.
- Strip group and strip realm—These parameters direct the way the security appliance processes the usernames it receives. They apply only to usernames received in the form user@realm. A realm is an administrative domain appended to a username with the @ delimiter (user@abc).

  When you specify the **strip-group** command, the security appliance selects the connection profile for user connections by obtaining the group name from the username presented by the VPN client. The security appliance then sends only the user part of the username for authorization/authentication. Otherwise (if disabled), the security appliance sends the entire username, including the realm.

  Strip-realm processing removes the realm from the username when sending the username to the authentication or authorization server. If the command is enabled, the security appliance sends only the user part of the username authorization/authentication. Otherwise, the security appliance sends the entire username.
- Authorization required—This parameter lets you require authorization before a user can connect, or turn off that requirement.
- Authorization DN attributes—This parameter specifies which Distinguished Name attributes to use when performing authorization.

IPSec Tunnel-Group Connection Parameters

IPSec parameters include the following:

- A client authentication method: preshared keys, certificates, or both.
  - For IKE connections based on preshared keys, this is the alphanumeric key itself (up to 128 characters long), associated with the connection policy.
  - Peer-ID validation requirement—This parameter specifies whether to require validating the identity of the peer using the peer's certificate.
- An extended hybrid authentication method: XAUTH and hybrid XAUTH.

  You use **isakmp ikev1-user-authentication** command to implement hybrid XAUTH authentication when you need to use digital certificates for security appliance authentication and a different, legacy method for remote VPN user authentication, such as RADIUS, TACACS+ or SecurID.
- ISAKMP (IKE) keepalive settings. This feature lets the security appliance monitor the continued presence of a remote peer and report its own presence to that peer. If the peer becomes unresponsive, the security appliance removes the connection. Enabling IKE keepalives prevents hung connections when the IKE peer loses connectivity.

  There are various forms of IKE keepalives. For this feature to work, both the security appliance and its remote peer must support a common form. This feature works with the following peers:
  - Cisco AnyConnect VPN Client
  - Cisco VPN Client (Release 3.0 and above)
  - Cisco VPN 3000 Client (Release 2.x)
  - Cisco VPN 3002 Hardware Client
  - Cisco VPN 3000 Series Concentrators
  - Cisco IOS software
  - Cisco Secure PIX Firewall

  Non-Cisco VPN clients do not support IKE keepalives.

  If you are configuring a group of mixed peers, and some of those peers support IKE keepalives and others do not, enable IKE keepalives for the entire group. The feature does not affect the peers that do not support it.

  If you disable IKE keepalives, connections with unresponsive peers remain active until they time out, so we recommend that you keep your idle timeout short. To change your idle timeout, see "Configuring Group Policies" section on page 32-38.

Note To reduce connectivity costs, disable IKE keepalives if this group includes any clients connecting via ISDN lines. ISDN connections normally disconnect if idle, but the IKE keepalive mechanism prevents connections from idling and therefore from disconnecting.

If you do disable IKE keepalives, the client disconnects only when either its IKE or IPSec keys expire. Failed traffic does not disconnect the tunnel with the Peer Timeout Profile values as it does when IKE keepalives are enabled.

Note If you have a LAN-to-LAN configuration using IKE main mode, make sure that the two peers have the same IKE keepalive configuration. Both peers must have IKE keepalives enabled or both peers must have it disabled.

- If you configure authentication using digital certificates, you can specify whether to send the entire certificate chain (which sends the peer the identity certificate and all issuing certificates) or just the issuing certificates (including the root certificate and any subordinate CA certificates).
- You can notify users who are using outdated versions of Windows client software that they need to update their client, and you can provide a mechanism for them to get the updated client version. For VPN 3002 hardware client users, you can trigger an automatic update. You can configure and change the client-update, either for all connection profiles or for particular connection profiles.
- If you configure authentication using digital certificates, you can specify the name of the trustpoint that identifies the certificate to send to the IKE peer.

Connection Profile Connection Parameters for Clientless SSL VPN Sessions

Table 32-1 provides a list of connection profile attributes that are specific to clientless SSL VPN. In addition to these attributes, you configure general connection profile attributes common to all VPN connections. For step-by-step information on configuring connection profiles, see "Configuring Connection Profiles for Clientless SSL VPN Sessions" in Chapter 32, "Configuring Connection Profiles, Group Policies, and Users."

Note In earlier releases, "connection profiles" were known as "tunnel groups." You configure a connection profile with tunnel-group commands. This chapter often uses these terms interchangeably.

Table 32-1 Connection Profile Attributes for Clientless SSL VPN

Command	Function
authentication	Sets the authentication method, AAA or certificate.
customization	Identifies the name of a previously defined customization to apply. Customizations determine the appearance of the windows that the user sees upon login. You configure the customization parameters as part of configuring clientless SSL VPN.
nbns-server	Identifies the name of the NetBIOS Name Service server (nbns-server) to use for CIFS name resolution.
group-alias	Specifies one or more alternate names by which the server can refer to a connection profile. At login, the user selects the group name from a dropdown menu.
group-url	Identifies one or more group URLs. If you configure this attribute, users coming in on a specified URL need not select a group at login.
dns-group	Identifies the DNS server group that specifies the DNS server name, domain name, name server, number of retries, and timeout values for a DNS server to use for a connection profile.

Table 32-1 Connection Profile Attributes for Clientless SSL VPN

Command	Function
hic-fail-group-policy	Specifies a VPN feature policy if you use the Cisco Secure Desktop Manager to set the Group-Based Policy attribute to "Use Failure Group-Policy" or "Use Success Group-Policy, if criteria match."
override-svc-download	Overrides downloading the group-policy or username attributes configured for downloading the AnyConnect VPN client to the remote user.
radius-reject-message	Enables the display of the RADIUS reject message on the login screen when authentication is rejected.

Configuring Connection Profiles

The following sections describe the contents and configuration of connection profiles:

- Maximum Connection Profiles, page 32-6
- Default IPSec Remote Access Connection Profile Configuration, page 32-7
- Specifying a Name and Type for the IPSec Remote Access Connection Profile, page 32-8
- Configuring IPSec Remote-Access Connection Profiles, page 32-8
- Configuring LAN-to-LAN Connection Profiles, page 32-16
- Configuring Connection Profiles for Clientless SSL VPN Sessions, page 32-19
- Customizing Login Windows for Users of Clientless SSL VPN sessions, page 32-27
- Configuring the Connection Profile for RADIUS/SDI Message Support for the AnyConnect Client, page 32-34

You can modify the default connection profiles, and you can configure a new connection profile as any of the three tunnel-group types. If you don't explicitly configure an attribute in a connection profile, that attribute gets its value from the default connection profile. The default connection-profile type is remote access. The subsequent parameters depend upon your choice of tunnel type. To see the current configured and default configuration of all your connection profiles, including the default connection profile, enter the **show running-config all tunnel-group** command.

Maximum Connection Profiles

The maximum number of connection profiles (tunnel groups) that an security appliance can support is a function of the maximum number of concurrent VPN sessions for the platform + 5. For example, an ASA5505 can support a maximum of 25 concurrent VPN sessions allowing for 30 tunnel groups (25+5). Attempting to add an additional tunnel group beyond the limit results in the following message: "ERROR: The limit of 30 configured tunnel groups has been reached"

Table Table 32-2specifies the maximum VPN sessions and connection profiles for each ASA platform.

Table 32-2 Maximum VPN Sessions and Connection Profiles Per ASA Platform

	5505 Base/ Security Plus	5510/Base/ Security Plus	5520	5540	5550
Maximum VPN Sessions	10/25	250	750	5000	5000
Maximum Connection Profiles	15/30	255	755	5005	5005

Default IPSec Remote Access Connection Profile Configuration

The contents of the default remote-access connection profile are as follows:

```
tunnel-group DefaultRAGroup type remote-access
tunnel-group DefaultRAGroup general-attributes
 no address-pool
 no ipv6-address-pool
 authentication-server-group LOCAL
 accounting-server-group RADIUS
 default-group-policy DfltGrpPolicy
 no dhcp-server
 no strip-realm
 no password-management
 no override-account-disable
 no strip-group
 no authorization-required
 authorization-dn-attributes CN OU
tunnel-group DefaultRAGroup webvpn-attributes
 hic-fail-group-policy DfltGrpPolicy
 customization DfltCustomization
 authentication aaa
 no override-svc-download
 no radius-reject-message
 dns-group DefaultDNS
tunnel-group DefaultRAGroup ipsec-attributes
 no pre-shared-key
 peer-id-validate req
 no chain
 no trust-point
 isakmp keepalive threshold 1500 retry 2
 no radius-sdi-xauth
 isakmp ikev1-user-authentication xauth
tunnel-group DefaultRAGroup ppp-attributes
 no authentication pap
 authentication chap
 authentication ms-chap-v1
 no authentication ms-chap-v2
 no authentication eap-proxy
```

Configuring IPSec Tunnel-Group General Attributes

The general attributes are common across more than one tunnel-group type. IPSec remote access and clientless SSL VPN tunnels share most of the same general attributes. IPSec LAN-to-LAN tunnels use a subset. Refer to the *Cisco Security Appliance Command Reference* for complete descriptions of all commands. The following sections describe, in order, how to configure IPSec remote-access connection profiles, IPSec LAN-to-LAN connection profiles, and clientless SSL VPN connection profiles.

Configuring IPSec Remote-Access Connection Profiles

Use an IPSec remote-access connection profile when setting up a connection between a remote client and a central-site security appliance, using a hardware or software client.To configure an IPSec remote-access connection profile, first configure the tunnel-group general attributes, then the IPSec remote-access attributes. An IPSec Remote Access VPN connection profile applies only to remote-access IPSec client connections. To configure an IPSec remote-access connection profile, see the following sections:

- Specifying a Name and Type for the IPSec Remote Access Connection Profile, page 32-8.
- Configuring IPSec Remote-Access Connection Profile General Attributes, page 32-8.
- Configuring IPSec Remote-Access Connection Profile IPSec Attributes, page 32-13.

Specifying a Name and Type for the IPSec Remote Access Connection Profile

Create the connection profile, specifying its name and type, by entering the **tunnel-group** command. For an IPSec remote-access tunnel, the type is **remote-access**

```
hostname(config)# tunnel-group tunnel_group_name type remote-access
hostname(config)#
```

For example, to create an IPSec remote-access connection profile named TunnelGroup1, enter the following command:

```
hostname(config)# tunnel-group TunnelGroup1 type remote-access
hostname(config)#
```

Configuring IPSec Remote-Access Connection Profile General Attributes

To configure or change the connection profile general attributes, specify the parameters in the following steps.

Step 1 To configure the general attributes, enter the **tunnel-group general-attributes** command, which enters tunnel-group general-attributes configuration mode. The prompt changes to indicate the change in mode.

```
hostname(config)# tunnel-group tunnel_group_name general-attributes
hostname(config-tunnel-general)#
```

Step 2 Specify the name of the authentication-server group, if any, to use. If you want to use the LOCAL database for authentication if the specified server group fails, append the keyword **LOCAL**:

```
hostname(config-tunnel-general)# authentication-server-group [(interface_name)] groupname
[LOCAL]
hostname(config-tunnel-general)#
```

The name of the authentication server group can be up to 16 characters long.

You can optionally configure interface-specific authentication by including the name of an interface after the group name. The interface name, which specifies where the IPSec tunnel terminates, must be enclosed in parentheses. The following command configures interface-specific authentication for the interface named test using the server named servergroup1 for authentication:

```
hostname(config-tunnel-general)# authentication-server-group (test) servergroup1
hostname(config-tunnel-general)#
```

Step 3 Specify the name of the authorization-server group, if any, to use. When you configure this value, users must exist in the authorization database to connect:

```
hostname(config-tunnel-general)# authorization-server-group groupname
hostname(config-tunnel-general)#
```

The name of the authorization server group can be up to 16 characters long. For example, the following command specifies the use of the authorization-server group FinGroup:

```
hostname(config-tunnel-general)# authorization-server-group FinGroup
hostname(config-tunnel-general)#
```

Step 4 Specify the name of the accounting-server group, if any, to use:

```
hostname(config-tunnel-general)# accounting-server-group groupname
hostname(config-tunnel-general)#
```

The name of the accounting server group can be up to 16 characters long. For example, the following command specifies the use of the accounting-server group named comptroller:

```
hostname(config-tunnel-general)# accounting-server-group comptroller
hostname(config-tunnel-general)#
```

Step 5 Specify the name of the default group policy:

```
hostname(config-tunnel-general)# default-group-policy policyname
hostname(config-tunnel-general)#
```

The name of the group policy can be up to 64 characters long. The following example sets DfltGrpPolicy as the name of the default group policy:

```
hostname(config-tunnel-general)# default-group-policy DfltGrpPolicy
hostname(config-tunnel-general)#
```

Step 6 Specify the names or IP addresses of the DHCP server (up to 10 servers), and the names of the DHCP address pools (up to 6 pools). The defaults are no DHCP server and no address pool. The **dhcp-server** command will allow you to configure the security appliance to send additional options to the specified DHCP servers when it is trying to get IP addresses for VPN clients. See the **dhcp-server** command in the *Cisco Security Appliance Command Reference* guide for more information.

```
hostname(config-tunnel-general)# dhcp-server server1 [...server10]
hostname(config-tunnel-general)# address-pool [(interface name)] address_pool1
[...address_pool6]
hostname(config-tunnel-general)#
```

Note If you specify an interface name, you must enclosed it within parentheses.

You configure address pools with the **ip local pool** command in global configuration mode.

Step 7 Specify the name of the NAC authentication server group, if you are using Network Admission Control, to identify the group of authentication servers to be used for Network Admission Control posture validation. Configure at least one Access Control Server to support NAC. Use the **aaa-server** command to name the ACS group. Then use the **nac-authentication-server-group** command, using the same name for the server group.

The following example identifies acs-group1 as the authentication server group to be used for NAC posture validation:

```
hostname(config-group-policy)# nac-authentication-server-group acs-group1
hostname(config-group-policy)
```

The following example inherits the authentication server group from the default remote access group.

```
hostname(config-group-policy)# no nac-authentication-server-group
hostname(config-group-policy)
```

Note NAC requires a Cisco Trust Agent on the remote host.

Step 8 Specify whether to strip the group or the realm from the username before passing it on to the AAA server. The default is not to strip either the group name or the realm.

```
hostname(config-tunnel-general)# strip-group
hostname(config-tunnel-general)# strip-realm
hostname(config-tunnel-general)#
```

A realm is an administrative domain. If you strip the realm, the security appliance uses the username and the group (if present) authentication. If you strip the group, the security appliance uses the username and the realm (if present) for authentication.Enter the **strip-realm** command to remove the realm qualifier, and use the strip-group command to remove the group qualilfier from the username during authentication. If you remove both qualifiers, authentication is based on the *username* alone. Otherwise, authentication is based on the full *username@realm* or *username<delimiter> group* string. You must specify **strip-realm** if your server is unable to parse delimiters.

Step 9 Optionally, if your server is a RADIUS, RADIUS with NT, or LDAP server, you can enable password management.

Note If you are using an LDAP directory server for authentication, password management is supported with the Sun Microsystems JAVA System Directory Server (formerly named the Sun ONE Directory Server) and the Microsoft Active Directory.

Sun—The DN configured on the security appliance to access a Sun directory server must be able to access the default password policy on that server. We recommend using the directory administrator, or a user with directory administrator privileges, as the DN. Alternatively, you can place an ACI on the default password policy.

Microsoft—You must configure LDAP over SSL to enable password management with Microsoft Active Directory.

See the "Setting the LDAP Server Type" section on page 14-13 for more information.

This feature, which is enabled by default, warns a user when the current password is about to expire. The default is to begin warning the user 14 days before expiration:

```
hostname(config-tunnel-general)# password-management
hostname(config-tunnel-general)#
```

If the server is an LDAP server, you can specify the number of days (0 through 180) before expiration to begin warning the user about the pending expiration:

```
hostname(config-tunnel-general)# password-management [password-expire in days n]
hostname(config-tunnel-general)#
```

Note The **password-management** command, entered in tunnel-group general-attributes configuration mode replaces the deprecated **radius-with-expiry** command that was formerly entered in tunnel-group ipsec-attributes mode.

When you configure the **password-management** command, the security appliance notifies the remote user at login that the user's current password is about to expire or has expired. The security appliance then offers the user the opportunity to change the password. If the current password has not yet expired, the user can still log in using that password. The security appliance ignores this command if RADIUS or LDAP authentication has not been configured.

Note that this does not change the number of days before the password expires, but rather, the number of days ahead of expiration that the security appliance starts warning the user that the password is about to expire.

If you do specify the **password-expire-in-days** keyword, you must also specify the number of days.

Specifying this command with the number of days set to 0 disables this command. The security appliance does not notify the user of the pending expiration, but the user can change the password after it expires.

See Configuring Microsoft Active Directory Settings for Password Management, page 32-27 for more information.



Note The security appliance, releases 7.1 and later, generally supports password management for the AnyConnect VPN Client, the Cisco IPSec VPN Client, the SSL VPN full-tunneling client, and Clientless connections when authenticating with LDAP or with any RADIUS connection that supports MS-CHAPv2. Password management is *not* supported for any of these connection types for Kerberos/AD (Windows password) or NT 4.0 Domain.

Some RADIUS servers that support MS-CHAP do not currently support MS-CHAPv2. The **password-management** command requires MS-CHAPv2, so please check with your vendor.

The RADIUS server (for example, Cisco ACS) could proxy the authentication request to another authentication server. However, from the security appliance perspective, it is talking only to a RADIUS server.

For LDAP, the method to change a password is proprietary for the different LDAP servers on the market. Currently, the security appliance implements the proprietary password management logic only for Microsoft Active Directory and Sun LDAP servers. Native LDAP requires an SSL connection. You must enable LDAP over SSL before attempting to do password management for LDAP. By default, LDAP uses port 636.

Step 10 Optionally, configure the ability to override an account-disabled indicator from a AAA server, by entering the **override-account-disable** command:

```
hostname(config-tunnel-general)# override-account-disable
hostname(config-tunnel-general)#
```



Note Allowing override-account-disable is a potential security risk.

Step 11 Specify the attribute or attributes to use in deriving a name for an authorization query from a certificate. This attribute specifies what part of the subject DN field to use as the username for authorization:

```
hostname(config-tunnel-general)# authorization-dn-attributes {primary-attribute
[secondary-attribute] | use-entire-name}
```

For example, the following command specifies the use of the CN attribute as the username for authorization:

```
hostname(config-tunnel-general)# authorization-dn-attributes CN
hostname(config-tunnel-general)#
```

The authorization-dn-attributes are **C** (Country), **CN** (Common Name), **DNQ** (DN qualifier), **EA** (E-mail Address), **GENQ** (Generational qualifier), **GN** (Given Name), **I** (Initials), **L** (Locality), **N** (Name), **O** (Organization), **OU** (Organizational Unit), **SER** (Serial Number), **SN** (Surname), **SP** (State/Province), **T** (Title), **UID** (User ID), and **UPN** (User Principal Name).

Step 12 Specify whether to require a successful authorization before allowing a user to connect. The default is not to require authorization.

```
hostname(config-tunnel-general)# authorization-required
hostname(config-tunnel-general)#
```

Enabling IPv6 VPN Access

The security appliance allows access to IPv6 resources over a public IPv4 connection (Windows XP SP2, Windows Vista, Mac OSX, and Linux only). If you want to configure IPv6 access, you must use the command-line interface to configure IPv6; ASDM does not support IPv6.

You enable IPv6 access using the **ipv6 enable** command as part of enabling SSL VPN connections. The following is an example for an IPv6 connection that enables IPv6 on the outside interface:

```
hostname(config)# interface GigabitEthernet0/0
hostname(config-if)# ipv6 enable
```

To enable IPV6 SSL VPN, do the following general actions:

1. Enable IPv6 on the outside interface.
2. Enable IPv6 and an IPv6 address on the inside interface.
3. Configure an IPv6 address local pool for client assigned IP Addresses.
4. Configure an IPv6 tunnel default gateway.

To implement this procedure, do the following steps:

Step 1 Configure Interfaces:

```
interface GigabitEthernet0/0
     nameif outside
     security-level 0
     ip address 192.168.0.1 255.255.255.0
     ipv6 enable          ; Needed for IPv6.
    !
interface GigabitEthernet0/1
     nameif inside
     security-level 100
     ip address 10.10.0.1 255.255.0.0
     ipv6 address 2001:DB8::1/32        ; Needed for IPv6.
     ipv6 enable          ; Needed for IPv6.
```

Step 2 Configure an 'ipv6 local pool' (used for IPv6 address assignment):

```
ipv6 local pool ipv6pool 2001:DB8:1:1::5/32 100     ; Use your IPv6 prefix here
```

Note You still need to configure an IPv4 address pool when using IPv6 (using the ip local pool command)

Step 3 Add the ipv6 address pool to your tunnel group policy (or group-policy):

```
tunnel-group YourTunGrp1 general-attributes  ipv6-address-pool ipv6pool
```

Note Again, you must also configure an IPv4 address pool here as well (using the 'address-pool' command).

Step 4 Configure an IPv6 tunnel default gateway:

```
ipv6 route inside ::/0 X:X:X:X::X tunneled
```

Configuring IPSec Remote-Access Connection Profile IPSec Attributes

To configure the IPSec attributes for a remote-access connection profile, do the following steps. The following description assumes that you have already created the IPSec remote-access connection profile. IPSec remote-access connection profiles have more attributes than IPSec LAN-to-LAN connection profiles:

Step 1 To specify the attributes of an IPSec remote-access tunnel-group, enter tunnel-group ipsec-attributes mode by entering the following command. The prompt changes to indicate the mode change:

```
hostname(config)# tunnel-group tunnel-group-name ipsec-attributes
hostname(config-tunnel-ipsec)#
```

This command enters tunnel-group ipsec-attributes configuration mode, in which you configure the remote-access tunnel-group IPSec attributes.

For example, the following command designates that the tunnel-group ipsec-attributes mode commands that follow pertain to the connection profile named TG1. Notice that the prompt changes to indicate that you are now in tunnel-group ipsec-attributes mode:

```
hostname(config)# tunnel-group TG1 type remote-access
hostname(config)# tunnel-group TG1 ipsec-attributes
hostname(config-tunnel-ipsec)#
```

Step 2 Specify the preshared key to support IKE connections based on preshared keys. For example, the following command specifies the preshared key xyzx to support IKE connections for an IPSec remote access connection profile:

```
hostname(config-tunnel-ipsec)# pre-shared-key xyzx
hostname(config-tunnel-ipsec)#
```

Step 3 Specify whether to validate the identity of the peer using the peer's certificate:

```
hostname(config-tunnel-ipsec)# peer-id-validate option
hostname(config-tunnel-ipsec)#
```

The available options are **req** (required), **cert** (if supported by certificate), and **nocheck** (do not check). The default is **req**.

For example, the following command specifies that peer-id validation is required:

```
hostname(config-tunnel-ipsec)# peer-id-validate req
hostname(config-tunnel-ipsec)#
```

Step 4 Specify whether to

Step 5 Specify whether to enable sending of a certificate chain. The following command includes the root certificate and any subordinate CA certificates in the transmission:

```
hostname(config-tunnel-ipsec)# chain
hostname(config-tunnel-ipsec)#
```

This attribute applies to all IPSec tunnel-group types.

Step 6 Specify the name of a trustpoint that identifies the certificate to be sent to the IKE peer:

```
hostname(config-tunnel-ipsec)# trust-point trust-point-name
hostname(config-tunnel-ipsec)#
```

The following command specifies mytrustpoint as the name of the certificate to be sent to the IKE peer:

```
hostname(config-ipsec)# trust-point mytrustpoint
```

Step 7 Specify the ISAKMP (IKE) keepalive threshold and the number of retries allowed.

```
hostname(config-tunnel-ipsec)# isakmp keepalive threshold <number> retry <number>
hostname(config-tunnel-ipsec)#
```

The **threshold** parameter specifies the number of seconds (10 through 3600) that the peer is allowed to idle before beginning keepalive monitoring. The **retry** parameter is the interval (2 through 10 seconds) between retries after a keepalive response has not been received. IKE keepalives are enabled by default. To disable IKE keepalives, enter the **no** form of the **isakmp** command:

For example, the following command sets the IKE keepalive threshold value to 15 seconds and sets the retry interval to 10 seconds:

```
hostname(config-tunnel-ipsec)# isakmp keepalive threshold 15 retry 10
hostname(config-tunnel-ipsec)#
```

The default value for the **threshold** parameter is 300 for remote-access and 10 for LAN-to-LAN, and the default value for the retry parameter is 2.

To specify that the central site (“head end”) should never initiate ISAKMP monitoring, enter the following command:

```
hostname(config-tunnel-ipsec)# isakmp keepalive threshold infinite
hostname(config-tunnel-ipsec)#
```

Step 8 Specify the ISAKMP hybrid authentication method, XAUTH or hybrid XAUTH.

You use **isakmp ikev1-user-authentication** command to implement hybrid XAUTH authentication when you need to use digital certificates for security appliance authentication and a different, legacy method for remote VPN user authentication, such as RADIUS, TACACS+ or SecurID. Hybrid XAUTH breaks phase 1 of IKE down into the following two steps, together called hybrid authentication:

a. The security appliance authenticates to the remote VPN user with standard public key techniques. This establishes an IKE security association that is unidirectionally authenticated.

b. An XAUTH exchange then authenticates the remote VPN user. This extended authentication can use one of the supported legacy authentication methods.



Note Before the authentication type can be set to hybrid, you must configure the authentication server, create a preshared key, and configure a trustpoint.

You can use the **isakmp ikev1-user-authentication** command with the optional **interface** parameter to specify a particular interface. When you omit the **interface** parameter, the command applies to all the interfaces and serves as a back-up when the per-interface command is not specified. When there are two **isakmp ikev1-user-authentication** commands specified for a connection profile, and one uses the **interface** parameter and one does not, the one specifying the interface takes precedence for that particular interface.

For example, the following commands enable hybrid XAUTH on the inside interface for a connection profile called example-group:

```
hostname(config)# tunnel-group example-group type remote-access
hostname(config)# tunnel-group example-group ipsec-attributes
hostname(config-tunnel-ipsec)# isakmp ikev1-user-authentication (inside) hybrid
hostname(config-tunnel-ipsec)#
```

Configuring IPSec Remote-Access Connection Profile PPP Attributes

To configure the Point-to-Point Protocol attributes for a remote-access connection profile, do the following steps. PPP attributes apply *only* to IPSec remote-access connection profiles. The following description assumes that you have already created the IPSec remote-access connection profile.

Step 1 Enter tunnel-group ppp-attributes configuration mode, in which you configure the remote-access tunnel-group PPP attributes, by entering the following command. The prompt changes to indicate the mode change:

```
hostname(config)# tunnel-group tunnel-group-name type remote-access
hostname(config)# tunnel-group tunnel-group-name ppp-attributes
hostname(config-tunnel-ppp)#
```

For example, the following command designates that the tunnel-group ppp-attributes mode commands that follow pertain to the connection profile named TG1. Notice that the prompt changes to indicate that you are now in tunnel-group ppp-attributes mode:

```
hostname(config)# tunnel-group TG1 type remote-access
hostname(config)# tunnel-group TG1 ppp-attributes
hostname(config-tunnel-ppp)#
```

Step 2 Specify whether to enable authentication using specific protocols for the PPP connection. The protocol value can be:

- pap—Enables the use of Password Authentication Protocol for the PPP connection.
- chap—Enables the use of Challenge Handshake Authentication Protocol for the PPP connection.
- ms-chap-v1 or ms-chap-v2—Enables the use of Microsoft Challenge Handshake Authentication Protocol, version 1 or version 2 for the PPP connection.
- eap—Enables the use of Extensible Authentication protocol for the PPP connection.

CHAP and MSCHAPv1 are enabled by default.

The syntax of this command is:

```
hostname(config-tunnel-ppp)# authentication protocol
hostname(config-tunnel-ppp)#
```

To disable authentication for a specific protocol, use the **no** form of the command:

```
hostname(config-tunnel-ppp)# no authentication protocol
hostname(config-tunnel-ppp)#
```

For example, the following command enables the use of the PAP protocol for a PPP connection.

```
hostname(config-tunnel-ppp)# authentication pap
hostname(config-tunnel-ppp)#
```

The following command enables the use of the MS-CHAP, version 2 protocol for a PPP connection:

```
hostname(config-tunnel-ppp)# authentication ms-chap-v2
hostname(config-tunnel-ppp)#
```

The following command enables the use of the EAP-PROXY protocol for a PPP connection:

```
hostname(config-tunnel-ppp)# authentication pap
hostname(config-tunnel-ppp)#
```

The following command disables the use of the MS-CHAP, version 1 protocol for a PPP connection:

```
hostname(config-tunnel-ppp)# no authentication ms-chap-v1
hostname(config-tunnel-ppp)#
```

Configuring LAN-to-LAN Connection Profiles

An IPSec LAN-to-LAN VPN connection profile applies only to LAN-to-LAN IPSec client connections. While many of the parameters that you configure are the same as for IPSec remote-access connection profiles, LAN-to-LAN tunnels have fewer parameters. To configure a LAN-to-LAN connection profile, follow the steps in this section.

Default LAN-to-LAN Connection Profile Configuration

The contents of the default LAN-to-LAN connection profile are as follows:

```
tunnel-group DefaultL2LGroup type ipsec-l2l
tunnel-group DefaultL2LGroup general-attributes
 no accounting-server-group
 default-group-policy DfltGrpPolicy
tunnel-group DefaultL2LGroup ipsec-attributes
 no pre-shared-key
 peer-id-validate req
 no chain
 no trust-point
 isakmp keepalive threshold 10 retry 2
```

LAN-to-LAN connection profiles have fewer parameters than remote-access connection profiles, and most of these are the same for both groups. For your convenience in configuring the connection, they are listed separately here. Any parameters that you do not explicitly configure inherit their values from the default connection profile.

Specifying a Name and Type for a LAN-to-LAN Connection Profile

To specify a name and a type for a connection profile, enter the **tunnel-group** command, as follows:

```
hostname(config)# tunnel-group tunnel_group_name type tunnel_type
```

For a LAN-to-LAN tunnel, the type is **ipsec-l2l**.; for example, to create the LAN-to-LAN connection profile named docs, enter the following command:

```
hostname(config)# tunnel-group docs type ipsec-l2l
hostname(config)#
```

Configuring LAN-to-LAN Connection Profile General Attributes

To configure the connection profile general attributes, do the following steps:

Step 1 Enter tunnel-group general-attributes mode by specifying the general-attributes keyword:

```
hostname(config)# tunnel-group_tunnel-group-name general-attributes
hostname(config-tunnel-general)#
```

The prompt changes to indicate that you are now in config-general mode, in which you configure the tunnel-group general attributes.

For example, for the connection profile named docs, enter the following command:

```
hostname(config)# tunnel-group_docs general-attributes
hostname(config-tunnel-general)#
```

Step 2 Specify the name of the accounting-server group, if any, to use:

```
hostname(config-tunnel-general)# accounting-server-group groupname
hostname(config-tunnel-general)#
```

For example, the following command specifies the use of the accounting-server group acctgserv1:

```
hostname(config-tunnel-general)# accounting-server-group acctgserv1
hostname(config-tunnel-general)#
```

Step 3 Specify the name of the default group policy:

```
hostname(config-tunnel-general)# default-group-policy policyname
hostname(config-tunnel-general)#
```

For example, the following command specifies that the name of the default group policy is MyPolicy:

```
hostname(config-tunnel-general)# default-group-policy MyPolicy
hostname(config-tunnel-general)#
```

Configuring LAN-to-LAN IPSec Attributes

To configure the IPSec attributes, do the following steps:

Step 1 To configure the tunnel-group IPSec attributes, enter tunnel-group ipsec-attributes configuration mode by entering the tunnel-group command with the IPSec-attributes keyword.

```
hostname(config)# tunnel-group tunnel-group-name ipsec-attributes
hostname(config-tunnel-ipsec)#
```

For example, the following command enters config-ipsec mode so you can configure the parameters for the connection profile named TG1:

```
hostname(config)# tunnel-group TG1 ipsec-attributes
hostname(config-tunnel-ipsec)#
```

The prompt changes to indicate that you are now in tunnel-group ipsec-attributes configuration mode.

Step 2 Specify the preshared key to support IKE connections based on preshared keys.

```
hostname(config-tunnel-ipsec)# pre-shared-key key
hostname(config-tunnel-ipsec)#
```

For example, the following command specifies the preshared key XYZX to support IKE connections for an IPSec LAN-to-LAN connection profile:

```
hostname(config-tunnel-ipsec)# pre-shared-key xyzx
hostname(config-tunnel-general)#
```

Step 3 Specify whether to validate the identity of the peer using the peer's certificate:

```
hostname(config-tunnel-ipsec)# peer-id-validate option
hostname(config-tunnel-ipsec)#
```

The available options are **req** (required), **cert (**if supported by certificate), and **nocheck (**do not check). The default is **req**. For example, the following command sets the peer-id-validate option to **nocheck**:

```
hostname(config-tunnel-ipsec)# peer-id-validate nocheck
hostname(config-tunnel-ipsec)#
```

Step 4 Specify whether to enable sending of a certificate chain. This action includes the root certificate and any subordinate CA certificates in the transmission:

```
hostname(config-tunnel-ipsec)# chain
hostname(config-tunnel-ipsec)#
```

You can apply this attribute to all tunnel-group types.

Step 5 Specify the name of a trustpoint that identifies the certificate to be sent to the IKE peer:

```
hostname(config-tunnel-ipsec)# trust-point trust-point-name
hostname(config-tunnel-ipsec)#
```

For example, the following command sets the trustpoint name to mytrustpoint:

```
hostname(config-tunnel-ipsec)# trust-point mytrustpoint
hostname(config-tunnel-ipsec)#
```

You can apply this attribute to all tunnel-group types.

Step 6 Specify the ISAKMP(IKE) keepalive threshold and the number of retries allowed. The **threshold** parameter specifies the number of seconds (10 through 3600) that the peer is allowed to idle before beginning keepalive monitoring. The **retry** parameter is the interval (2 through 10 seconds) between retries after a keepalive response has not been received. IKE keepalives are enabled by default. To disable IKE keepalives, enter the **no** form of the **isakmp** command:

```
hostname(config)# isakmp keepalive threshold <number> retry <number>
hostname(config-tunnel-ipsec)#
```

For example, the following command sets the ISAKMP keepalive threshold to 15 seconds and sets the retry interval to 10 seconds.:

```
hostname(config-tunnel-ipsec)# isakmp keepalive threshold 15 retry 10
hostname(config-tunnel-ipsec)#
```

The default value for the **threshold** parameter for LAN-to-LAN is 10, and the default value for the retry parameter is 2.

To specify that the central site ("head end") should never initiate ISAKMP monitoring, enter the following command:

```
hostname(config-tunnel-ipsec)# isakmp keepalive threshold infinite
hostname(config-tunnel-ipsec)#
```

Step 7 Specify the ISAKMP hybrid authentication method, XAUTH or hybrid XAUTH.

You use **isakmp ikev1-user-authentication** command to implement hybrid XAUTH authentication when you need to use digital certificates for security appliance authentication and a different, legacy method for remote VPN user authentication, such as RADIUS, TACACS+ or SecurID. Hybrid XAUTH breaks phase 1 of IKE down into the following two steps, together called hybrid authentication:

a. The security appliance authenticates to the remote VPN user with standard public key techniques. This establishes an IKE security association that is unidirectionally authenticated.

b. An XAUTH exchange then authenticates the remote VPN user. This extended authentication can use one of the supported legacy authentication methods.

Note Before the authentication type can be set to hybrid, you must configure the authentication server, create a preshared key, and configure a trustpoint.

You can use the **isakmp ikev1-user-authentication** command with the optional **interface** parameter to specify a particular interface. When you omit the **interface** parameter, the command applies to all the interfaces and serves as a back-up when the per-interface command is not specified. When there are two **isakmp ikev1-user-authentication** commands specified for a connection profile, and one uses the **interface** parameter and one does not, the one specifying the interface takes precedence for that particular interface.

For example, the following commands enable hybrid XAUTH on the inside interface for a connection profile called example-group:

```
hostname(config)# tunnel-group example-group type remote-access
hostname(config)# tunnel-group example-group ipsec-attributes
hostname(config-tunnel-ipsec)# isakmp ikev1-user-authentication (inside) hybrid
hostname(config-tunnel-ipsec)#
```

Configuring Connection Profiles for Clientless SSL VPN Sessions

The tunnel-group general attributes for clientless SSL VPN connection profiles are the same as those for IPSec remote-access connection profiles, except that the tunnel-group type is webvpn and the **strip-group** and **strip-realm** commands do not apply. You define the attribute specific to clientless SSL VPN separately. The following sections describe how to configure clientless SSL VPN connection profiles.

Specifying a Connection Profile Name and Type for Clientless SSL VPN Sessions

Create the connection profile, specifying its name and type by entering the **tunnel-group** command in global configuration mode. For an IPSec remote-access tunnel, the type is **webvpn**

```
hostname(config)# tunnel-group tunnel_group_name type webvpn
hostname(config)#
```

For example, to create a clientless SSL VPN tunnel-group named TunnelGroup3, enter the following command:

```
hostname(config)# tunnel-group TunnelGroup3 type webvpn
hostname(config)#
```

Configuring General Tunnel-Group Attributes for Clientless SSL VPN Sessions

To configure or change the connection profile general attributes, specify the parameters in the following steps.

Step 1 To configure the general attributes, enter **tunnel-group general-attributes** command, which enters tunnel-group general-attributes configuration mode. Note that the prompt changes:

```
hostname(config)# tunnel-group tunnel_group_name general-attributes
hostname(config-tunnel-general)#
```

To configure the general attributes for TunnelGroup3, created in the previous section, enter the following command:

```
hostname(config)# tunnel-group TunnelGroup3 general-attributes
hostname(config-tunnel-general)#
```

Step 2 Specify the name of the authentication-server group, if any, to use. If you want to use the LOCAL database for authentication if the specified server group fails, append the keyword LOCAL:

```
hostname(config-tunnel-general)# authentication-server-group groupname [LOCAL]
hostname(config-tunnel-general)#
```

For example, to configure the authentication server group named test, and to provide fallback to the LOCAL server if the authentication server group fails, enter the following command:

```
hostname(config-tunnel-general)# authentication-server-group test LOCAL
hostname(config-tunnel-general)#
```

The authentication-server-group name identifies a previously configured authentication server or group of servers. Use the **aaa-server** command to configure authentication servers. The maximum length of the group tag is 16 characters.

You can also configure interface-specific authentication by including the name of an interface in parentheses before the group name. The following interfaces are available by default:

- inside—Name of interface GigabitEthernet0/1
- outside— Name of interface GigabitEthernet0/0

Other interfaces you have configured (using the **interface** command) are also available. The following command configures interface-specific authentication for the interface named outside using the server servergroup1 for authentication:

```
hostname(config-tunnel-general)# authentication-server-group (outside) servergroup1
hostname(config-tunnel-general)#
```

Step 3 Optionally, specify the name of the authorization-server group, if any, to use. If you are not using authorization, go to Step 6. When you configure this value, users must exist in the authorization database to connect:

```
hostname(config-tunnel-general)# authorization-server-group groupname
hostname(config-tunnel-general)#
```

Use the **aaa-server** command to configure authorization servers. The maximum length of the group tag is 16 characters.

For example, the following command specifies the use of the authorization-server group FinGroup:

```
hostname(config-tunnel-general)# authorization-server-group FinGroup
hostname(config-tunnel-general)#
```

Step 4 Specify whether to require a successful authorization before allowing a user to connect. The default is not to require authorization.

```
hostname(config-tunnel-general)# authorization-required
hostname(config-tunnel-general)#
```

Step 5 Specify the attribute or attributes to use in deriving a name for an authorization query from a certificate. This attribute specifies what part of the subject DN field to use as the username for authorization:

```
hostname(config-tunnel-general)# authorization-dn-attributes {primary-attribute
[secondary-attribute] | use-entire-name}
```

For example, the following command specifies the use of the CN attribute as the username for authorization:

```
hostname(config-tunnel-general)# authorization-dn-attributes CN
hostname(config-tunnel-general)#
```

The authorization-dn-attributes are **C** (Country), **CN** (Common Name), **DNQ** (DN qualifier), **EA** (E-mail Address), **GENQ** (Generational qualifier), **GN** (Given Name), **I** (Initials), **L** (Locality), **N** (Name), **O** (Organization), **OU** (Organizational Unit), **SER** (Serial Number), **SN** (Surname), **SP** (State/Province), **T** (Title), **UID** (User ID), and **UPN** (User Principal Name).

Step 6 Optionally, specify the name of the accounting-server group, if any, to use. If you are not using accounting, go to Step 7. Use the **aaa-server** command to configure accounting servers. The maximum length of the group tag is 16 characters.:

```
hostname(config-tunnel-general)# accounting-server-group groupname
hostname(config-tunnel-general)#
```

For example, the following command specifies the use of the accounting-server group comptroller:

```
hostname(config-tunnel-general)# accounting-server-group comptroller
hostname(config-tunnel-general)#
```

Step 7 Optionally, specify the name of the default group policy. The default value is DfltGrpPolicy:

```
hostname(config-tunnel-general)# default-group-policy policyname
hostname(config-tunnel-general)#
```

The following example sets MyDfltGrpPolicy as the name of the default group policy:

```
hostname(config-tunnel-general)# default-group-policy MyDfltGrpPolicy
hostname(config-tunnel-general)#
```

Step 8 Optionally, specify the name or IP address of the DHCP server (up to 10 servers), and the names of the DHCP address pools (up to 6 pools). Separate the list items with spaces. The defaults are no DHCP server and no address pool.

```
hostname(config-tunnel-general)# dhcp-server server1 [...server10]
hostname(config-tunnel-general)# address-pool [(interface name)] address_pool1
[...address_pool6]
hostname(config-tunnel-general)#
```

Note The interface name must be enclosed in parentheses.

You configure address pools with the **ip local pool** command in global configuration mode. See Chapter 33, "Configuring IP Addresses for VPNs" for information about configuring address pools.

Step 9 Optionally, if your server is a RADIUS, RADIUS with NT, or LDAP server, you can enable password management.

Note If you are using an LDAP directory server for authentication, password management is supported with the Sun Microsystems JAVA System Directory Server (formerly named the Sun ONE Directory Server) and the Microsoft Active Directory.

- Sun—The DN configured on the security appliance to access a Sun directory server must be able to access the default password policy on that server. We recommend using the directory administrator, or a user with directory administrator privileges, as the DN. Alternatively, you can place an ACI on the default password policy.
- Microsoft—You must configure LDAP over SSL to enable password management with Microsoft Active Directory.

See the "Setting the LDAP Server Type" section on page 14-13 for more information.

This feature, which is enabled by default, warns a user when the current password is about to expire. The default is to begin warning the user 14 days before expiration:

```
hostname(config-tunnel-general)# password-management
hostname(config-tunnel-general)#
```

If the server is an LDAP server, you can specify the number of days (0 through 180) before expiration to begin warning the user about the pending expiration:

```
hostname(config-tunnel-general)# password-management [password-expire in days n]
hostname(config-tunnel-general)#
```

Note The **password-management** command, entered in tunnel-group general-attributes configuration mode replaces the deprecated **radius-with-expiry** command that was formerly entered in tunnel-group ipsec-attributes mode.

When you configure this command, the security appliance notifies the remote user at login that the user's current password is about to expire or has expired. The security appliance then offers the user the opportunity to change the password. If the current password has not yet expired, the user can still log in using that password. The security appliance ignores this command if RADIUS or LDAP authentication has not been configured.

Note that this does not change the number of days before the password expires, but rather, the number of days ahead of expiration that the security appliance starts warning the user that the password is about to expire.

If you do specify the **password-expire-in-days** keyword, you must also specify the number of days.

See Configuring Microsoft Active Directory Settings for Password Management, page 32-27 for more information.

Step 10 Specifying this command with the number of days set to 0 disables this command. The security appliance does not notify the user of the pending expiration, but the user can change the password after it expires.Optionally, configure the ability to override an account-disabled indicator from the AAA server, by entering the **override-account-disable** command:

```
hostname(config-tunnel-general)# override-account-disable
hostname(config-tunnel-general)#
```

Note Allowing override account-disabled is a potential security risk.

Configuring Tunnel-Group Attributes for Clientless SSL VPN Sessions

To configure the parameters specific to a clientless SSL VPN connection profile, follow the steps in this section. Clientless SSL VPN was formerly known as WebVPN, and you configure these attributes in tunnel-group webvpn-attributes mode.

Step 1 To specify the attributes of a clientless SSL VPN tunnel-group, enter tunnel-group webvpn-attributes mode by entering the following command. The prompt changes to indicate the mode change:

```
hostname(config)# tunnel-group tunnel-group-name webvpn-attributes
hostname(config-tunnel-ipsec)#
```

For example, to specify the webvpn-attributes for the clientless SSL VPN tunnel-group named sales, enter the following command:

```
hostname(config)# tunnel-group sales webvpn-attributes
hostname(config-tunnel-webvpn)#
```

Step 2 To specify the authentication method to use: AAA, digital certificates, or both, enter the **authentication** command. You can specify either aaa or certificate or both, in any order.

```
hostname(config-tunnel-webvpn)# authentication authentication_method
hostname(config-tunnel-webvpn)#
```

For example, The following command allows both AAA and certificate authentication:

```
hostname(config-tunnel-webvpn)# authentication aaa certificate
hostname(config-tunnel-webvpn)#
```

Applying Customization

Customizations determine the appearance of the windows that the user sees upon login. You configure the customization parameters as part of configuring clientless SSL VPN.

To apply a previously defined web-page customization to change the look-and-feel of the web page that the user sees at login, enter the customization command in username webvpn configuration mode:

```
hostname(config-username-webvpn)# customization {none | value customization_name}
hostname(config-username-webvpn)#
```

For example, to use the customization named blueborder, enter the following command:

```
hostname(config-username-webvpn)# customization value blueborder
hostname(config-username-webvpn)#
```

You configure the customization itself by entering the **customization** command in webvpn mode.

The following example shows a command sequence that first establishes a customization named “123” that defines a password prompt. The example then defines a clientless SSL VPN tunnel-group named “test” and uses the **customization** command to specify the use of the customization named “123”:

```
hostname(config)# webvpn
hostname(config-webvpn)# customization 123
hostname(config-webvpn-custom)# password-prompt Enter password
```

```
hostname(config-webvpn)# exit
hostname(config)# tunnel-group test type webvpn
hostname(config)# tunnel-group test webvpn-attributes
hostname(config-tunnel-webvpn)# customization value 123
hostname(config-tunnel-webvpn)#
```

Step 3 The security appliance queries NetBIOS name servers to map NetBIOS names to IP addresses. Clientless SSL VPN requires NetBIOS to access or share files on remote systems. Clientless SSL VPN uses NetBIOS and the CIFS protocol to access or share files on remote systems. When you attempt a file-sharing connection to a Windows computer by using its computer name, the file server you specify corresponds to a specific NetBIOS name that identifies a resource on the network.

To make the NBNS function operational, you must configure at least one NetBIOS server (host). You can configure up to three NBNS servers for redundancy. The security appliance uses the first server on the list for NetBIOS/CIFS name resolution. If the query fails, it uses the next server.

To specify the name of the NBNS (NetBIOS Name Service) server to use for CIFS name resolution, use the **nbns-server** command. You can enter up to three server entries. The first server you configure is the primary server, and the others are backups, for redundancy. You can also specify whether this is a master browser (rather than just a WINS server), the timeout interval, and the number of retries. A WINS server or a master browser is typically on the same network as the security appliance, or reachable from that network. You must specify the timeout interval before the number of retries:

```
hostname(config-tunnel-webvpn)# nbns-server {host-name | IP_address} [master]
[timeout seconds] [retry number]
hostname(config-tunnel-webvpn)#
```

For example, to configure the server named nbnsprimary as the primary server and the server 192.168.2.2 as the secondary server, each allowing three retries and having a 5-second timeout, enter the following command:

```
hostname(config)# name 192.168.2.1 nbnsprimary
hostname(config-tunnel-webvpn)# nbns-server nbnsprimary master timeout 5 retry 3
hostname(config-tunnel-webvpn)# nbns-server 192.168.2.2 timeout 5 retry 3
hostname(config-tunnel-webvpn)#
```

The timeout interval can range from 1 through 30 seconds (default 2), and the number of retries can be in the range 0 through 10 (default 2).

The **nbns-server** command in tunnel-group webvpn-attributes configuration mode replaces the deprecated **nbns-server** command in webvpn configuration mode.

Step 4 To specify alternative names for the group, use the **group-alias** command. Specifying the group alias creates one or more alternate names by which the user can refer to a tunnel-group. The group alias that you specify here appears in the drop-down list on the user's login page. Each group can have multiple aliases or no alias, each specified in separate commands. This feature is useful when the same group is known by several common names, such as "Devtest" and "QA".

For each group alias, enter a **group-alias** command. Each alias is enabled by default. You can optionally explicitly enable or disable each alias:

```
hostname(config-tunnel-webvpn)# group-alias alias [enable | disable]
hostname(config-tunnel-webvpn)#
```

For example, to enable the aliases QA and Devtest for a tunnel-group named QA, enter the following commands:

```
hostname(config-tunnel-webvpn)# group-alias QA enable
hostname(config-tunnel-webvpn)# group-alias Devtest enable
hostname(config-tunnel-webvpn)#
```

Note The webvpn tunnel-group-list must be enabled for the (dropdown) group list to appear.

Step 5 To specify incoming URLs or IP addresses for the group, use the **group-url** command. Specifying a group URL or IP address eliminates the need for the user to select a group at login. When a user logs in, the security appliance looks for the user's incoming URL or address in the tunnel-group-policy table. If it finds the URL or address and if group-url is enabled in the connection profile, then the security appliance automatically selects the associated connection profile and presents the user with only the username and password fields in the login window. This simplifies the user interface and has the added advantage of never exposing the list of groups to the user. The login window that the user sees uses the customizations configured for that connection profile.

If the URL or address is disabled and group-alias is configured, then the dropdown list of groups is also displayed, and the user must make a selection.

You can configure multiple URLs or addresses (or none) for a group. Each URL or address can be enabled or disabled individually. You must use a separate **group-url** command for each URL or address specified. You must specify the entire URL or address, including either the http or https protocol.

You cannot associate the same URL or address with multiple groups. The security appliance verifies the uniqueness of the URL or address before accepting the URL or address for a connection profile.

For each group URL or address, enter a **group-url** command. You can optionally explicitly enable (the default) or disable each URL or alias:

```
hostname(config-tunnel-webvpn)# group-url url [enable | disable]
hostname(config-tunnel-webvpn)#
```

For example, to enable the group URLs http://www.cisco.com and http://192.168.10.10 for the tunnel-group named RadiusServer, enter the following commands:

```
hostname(config)# tunnel-group RadiusServer type webvpn
hostname(config)# tunnel-group RadiusServer general-attributes
hostname(config-tunnel-general)# authentication server-group RADIUS
hostname(config-tunnel-general)# accounting-server-group RADIUS
hostname(config-tunnel-general)# tunnel-group RadiusServer webvpn-attributes
hostname(config-tunnel-webvpn)# group-alias "Cisco Remote Access" enable
hostname(config-tunnel-webvpn)# group-url http://www.cisco.com enable
hostname(config-tunnel-webvpn)# group-url http://192.168.10.10 enable
hostname(config-tunnel-webvpn)#
```

For a more extensive example, see Customizing Login Windows for Users of Clientless SSL VPN sessions, page 32-27.

Step 6 To specify the DNS server to use for a connection profile for clientless SSL VPN sessions, enter the **dns-group** command. The default value is DefaultDNS:

```
hostname(config-tunnel-webvpn)# dns-group {hostname | ip_address}
hostname(config-tunnel-webvpn)#
```

The dns-group command resolves the hostname to the appropriate DNS server for the connection profile. For example, to specify the use of the DNS server named server1, enter the following command:

```
hostname(config)# name 10.10.10.1 server1
hostname(config-tunnel-webvpn)# dns-group server1
hostname(config-tunnel-webvpn)#
```

Step 7 (Optional) To specify a VPN feature policy if you use the Cisco Secure Desktop Manager to set the Group-Based Policy attribute to "Use Failure Group-Policy" or "Use Success Group-Policy, if criteria match," use the **hic-fail-group-policy** command. The default value is DfltGrpPolicy.

```
hostname(config-tunnel-webvpn)# hic-fail-group-policy name
hostname(config-tunnel-webvpn)#
```

Name is the name of a group policy created for a connection profile for clientless SSL VPN sessions.

This policy is an alternative group policy to differentiate access rights for the following CSD clients:

- Clients that match a CSD location entry set to "Use Failure Group-Policy."
- Clients that match a CSD location entry set to "Use Success Group-Policy, if criteria match," and then fail to match the configured Group-Based Policy criteria. For more information, see the *Cisco Secure Desktop Configuration Guide for Cisco ASA 5500 Series Administrators*.

The following example specifies an alternative group policy named group2:

```
hostname(config-tunnel-webvpn)# hic-fail-group-policy group2
hostname(config-tunnel-webvpn)#
```

Note The security appliance does not use this attribute if you set the VPN feature policy to "Always use Success Group-Policy."

For more information, see the *Cisco Secure Desktop Configuration Guide for Cisco ASA 5500 Series Administration Guide*.

Step 8 (Optional) To specify whether to override the group policy or username attributes configuration for downloading an AnyConnect or SSL VPN client, use the override-svc-download command. This feature is disabled by default.

The security appliance allows clientless, AnyConnect, or SSL VPN client connections for remote users based on whether clientless and/or SSL VPN is enabled in the group policy or username attributes with the **vpn-tunnel-protocol** command. The **svc ask** command further modifies the client user experience by prompting the user to download the client or return to the WebVPN home page.

However, you might want clientless users logging in under specific tunnel groups to not experience delays waiting for the download prompt to expire before being presented with the clientless SSL VPN home page. You can prevent delays for these users at the connection profile level with the **override-svc-download** command. This command causes users logging through a connection profile to be immediately presented with the clientless SSL VPN home page regardless of the **vpn-tunnel-protocol** or **svc ask** command settings.

In the following example, the you enter tunnel-group webvpn attributes configuration mode for the connection profile *engineering* and enable the connection profile to override the group policy and username attribute settings for client download prompts:

```
hostname(config)# tunnel-group engineering webvpn-attributes
hostname(config-tunnel-webvpn)# override-svc-download
```

Step 9 (Optional) To enable the display of a RADIUS reject message on the login screen when authentication is rejected, use the **radius-eject-message** command:

The following example enables the display of a RADIUS rejection message for the connection profile named engineering:

```
hostname(config)# tunnel-group engineering webvpn-attributes
hostname(config-tunnel-webvpn)# radius-reject-message
```

Customizing Login Windows for Users of Clientless SSL VPN sessions

You can set up different login windows for different groups by using a combination of customization profiles and connection profiles. For example, assuming that you had created a customization profile called salesgui, you can create a connection profile for clientless SSL VPN sessions called sales that uses that customization profile, as the following example shows:

Step 1 In webvpn mode, define a customization for clientless SSL VPN access, in this case named salesgui and change the default logo to mycompanylogo.gif. You must have previously loaded mycompanylogo.gif onto the flash memory of the security appliance and saved the configuration. See "Chapter 39, "Configuring Clientless SSL VPN"" for details.

```
hostname# webvpn
hostname (config-webvpn)# customization value salesgui
hostname(config-webvpn-custom)# logo file disk0:\mycompanylogo.gif
hostname(config-webvpn-custom)#
```

Step 2 In global configuration mode, set up a username and associate with it the customization for clientless SSL VPN that you've just defined:

```
hostname# username seller attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# customization value salesgui
hostname(config-username-webvpn)# exit
hostname(config-username)# exit
hostname#
```

Step 3 In global configuration mode, create a tunnel-group for clientless SSL VPN sessions named sales:

```
hostname# tunnel-group sales type webvpn
hostname(config-tunnel-webvpn)#
```

Step 4 Specify that you want to use the salesgui customization for this connection profile:

```
hostname# tunnel-group sales webvpn-attributes
hostname(config-tunnel-webvpn)# customization salesgui
```

Step 5 Set the group URL to the address that the user enters into the browser to log in to the security appliance; for example, if the security appliance has the IP address 192.168.3.3, set the group URL to https://192.168.3.3:

```
hostname(config-tunnel-webvpn)# group-url https://192.168.3.3.
hostname(config-tunnel-webvpn)#
```

If a port number is required for a successful login, include the port number, preceded by a colon. The security appliance maps this URL to the sales connection profile and applies the salesgui customization profile to the login screen that the user sees upon logging in to https://192.168.3.3.

Configuring Microsoft Active Directory Settings for Password Management

Note If you are using an LDAP directory server for authentication, password management is supported with the Sun Microsystems JAVA System Directory Server (formerly named the Sun ONE Directory Server) and the Microsoft Active Directory.

- Sun—The DN configured on the security appliance to access a Sun directory server must be able to access the default password policy on that server. We recommend using the directory administrator, or a user with directory administrator privileges, as the DN. Alternatively, you can place an ACI on the default password policy.
- Microsoft—You must configure LDAP over SSL to enable password management with Microsoft Active Directory.

See the "Setting the LDAP Server Type" section on page 14-13 for more information.

To use password management with Microsoft Active Directory, you must set certain Active Directory parameters as well as configuring password management on the security appliance. This section describes the Active Directory settings associated with various password management actions. These descriptions assume that you have also enabled password management on the security appliance and configured the corresponding password management attributes. The specific steps in the following sections refer to Active Directory terminology under Windows 2000.

- Using Active Directory to Force the User to Change Password at Next Logon, page 32-28.
- Using Active Directory to Specify Maximum Password Age, page 32-30.
- Using Active Directory to Override an Account Disabled AAA Indicator, page 32-31
- Using Active Directory to Enforce Password Complexity, page 32-33.

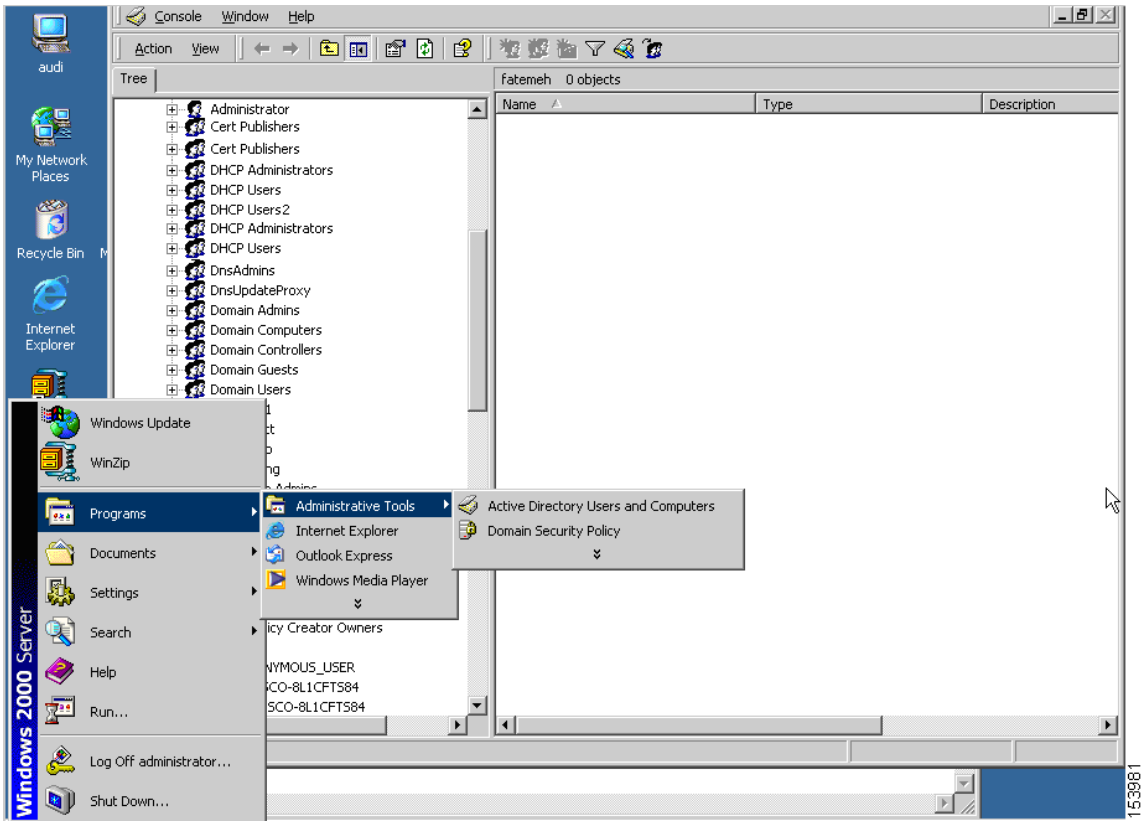
The following sections assume that you are using an LDAP directory server for authentication.

Using Active Directory to Force the User to Change Password at Next Logon

To force a user to change the user password at the next logon, specify the **password-management** command in tunnel-group general-attributes configuration mode on the security appliance and do the following steps under Active Directory:

Step 1 Select to Start > Programs > Administrative Tools > Active Directory Users and Computers (Figure 32-1).

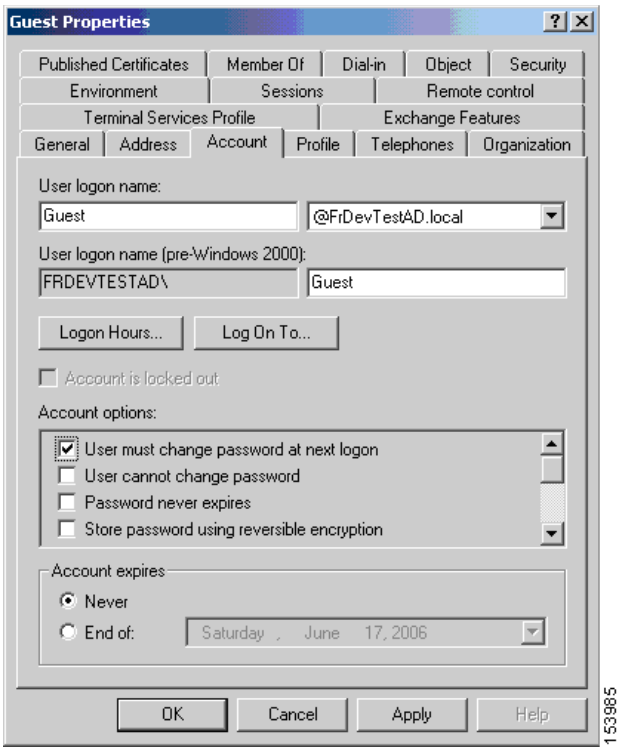
Figure 32-1 Active Directory—Administrative Tools Menu



Step 2 Right-click Username > Properties > Account.

Step 3 Check the check box for User must change password at next logon (Figure 32-2).

Figure 32-2 Active Directory—User Must Change Password at Next Logon



The next time this user logs on, the security appliance displays the following prompt: "New password required. Password change required. You must enter a new password with a minimum length *n* to continue." You can set the minimum required password length, *n*, as part of the Active Directory configuration at Start > Programs > Administrative Tools > Domain Security Policy > Windows Settings > Security Settings > Account Policies > Password Policy. Select Minimum password length.

Using Active Directory to Specify Maximum Password Age

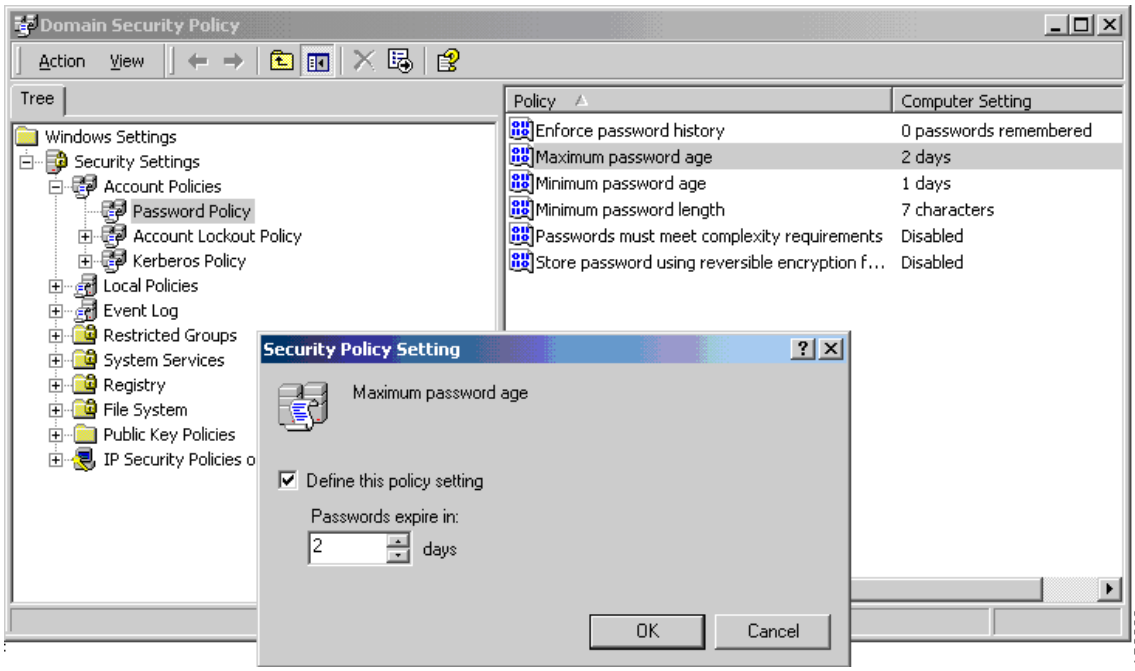
To enhance security, you can specify that passwords expire after a certain number of days. To specify a maximum password age for a user password, specify the **password-management** command in tunnel-group general-attributes configuration mode on the security appliance and do the following steps under Active Directory:

Step 1 Select Start > Programs > Administrative Tools > Domain Security Policy > Windows Settings > Security Settings > Account Policies > Password Policy.

Step 2 Double-click Maximum password age. This opens the Security Policy Setting dialog box.

Step 3 Check the Define this policy setting check box and specify the maximum password age, in days, that you want to allow.

Figure 32-3 Active Directory—Maximum Password Age



Note The **radius-with-expiry** command, formerly configured as part of tunnel-group remote-access configuration to perform the password age function, is deprecated. The **password-management** command, entered in tunnel-group general-attributes mode, replaces it.

Using Active Directory to Override an Account Disabled AAA Indicator

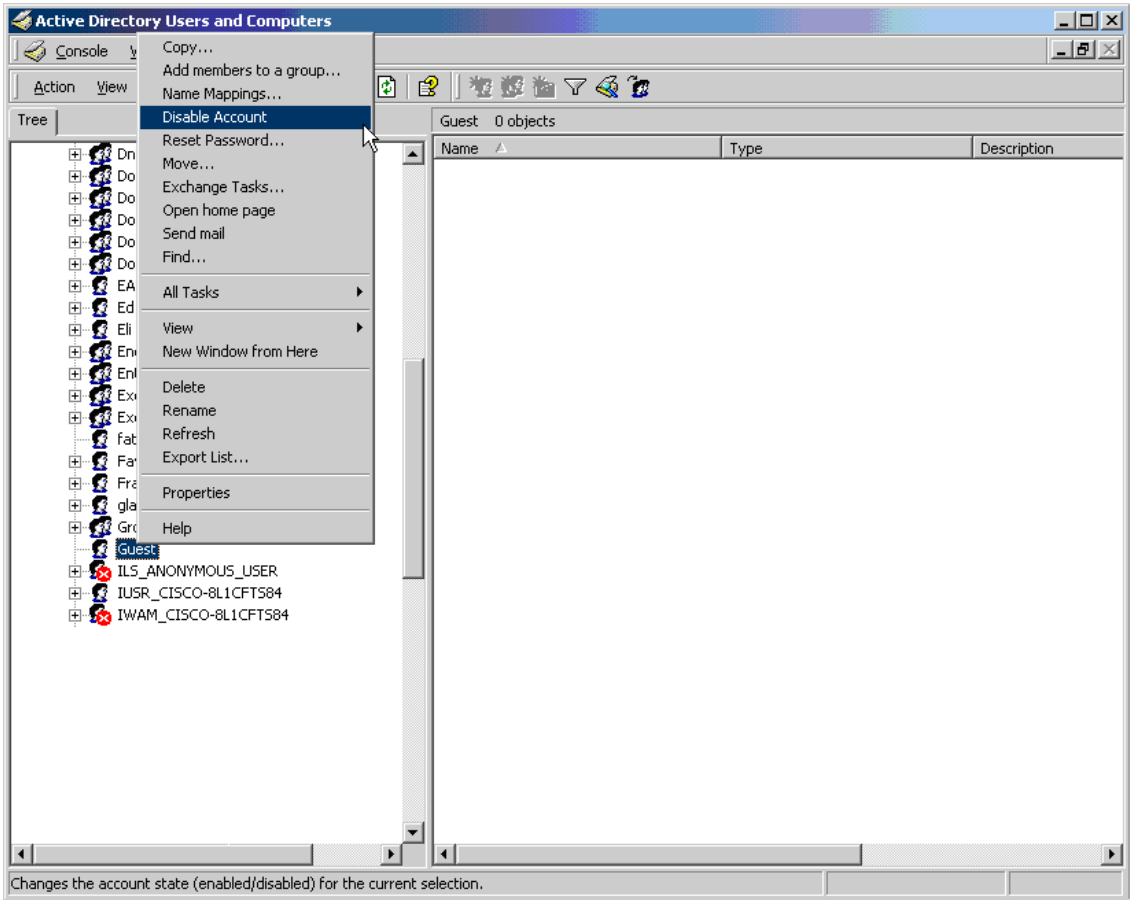
To override an account-disabled indication from a AAA server, specify the **override-account-disable** command in tunnel-group general-attributes configuration mode on thesecurity appliance and do the following steps under Active Directory:

Note Allowing override account-disabled is a potential security risk.

Step 1 Select Start > Programs > Administrative Tools > Active Directory Users and Computers.

Step 2 Right-click Username > Properties > Account and select Disable Account from the menu.

Figure 32-4 Active Directory—Override Account Disabled



The user should be able to log on successfully, even though a AAA server provides an account-disabled indicator.

Using Active Directory to Enforce Minimum Password Length

To enforce a minimum length for passwords, specify the **password-management** command in tunnel-group general-attributes configuration mode on the security appliance and do the following steps under Active Directory:

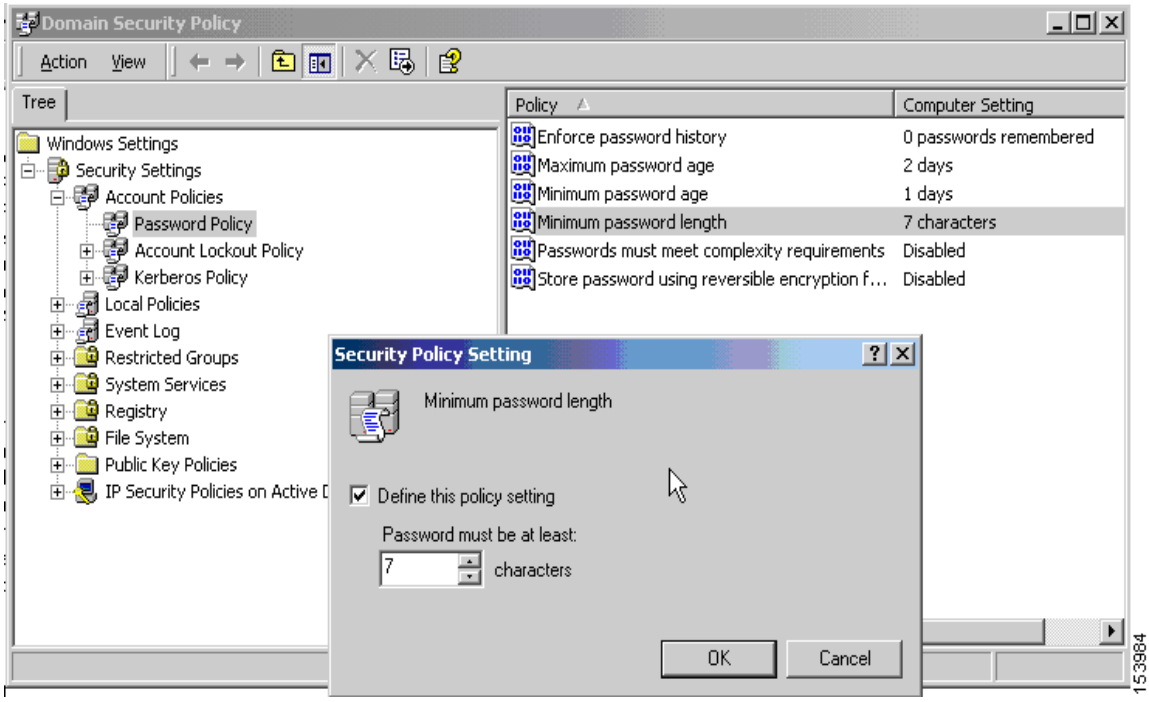
Step 1 Select Start > Programs > Administrative Tools > Domain Security Policy.

Step 2 Select Windows Settings > Security Settings > Account Policies > Password Policy.

Step 3 Double-click Minimum Password Length. This opens the Security Policy Setting dialog box.

Step 4 Check the Define this policy setting check box and specify the minimum number of characters that the password must contain.

Figure 32-5 Active Directory—Minimum Password Length



Using Active Directory to Enforce Password Complexity

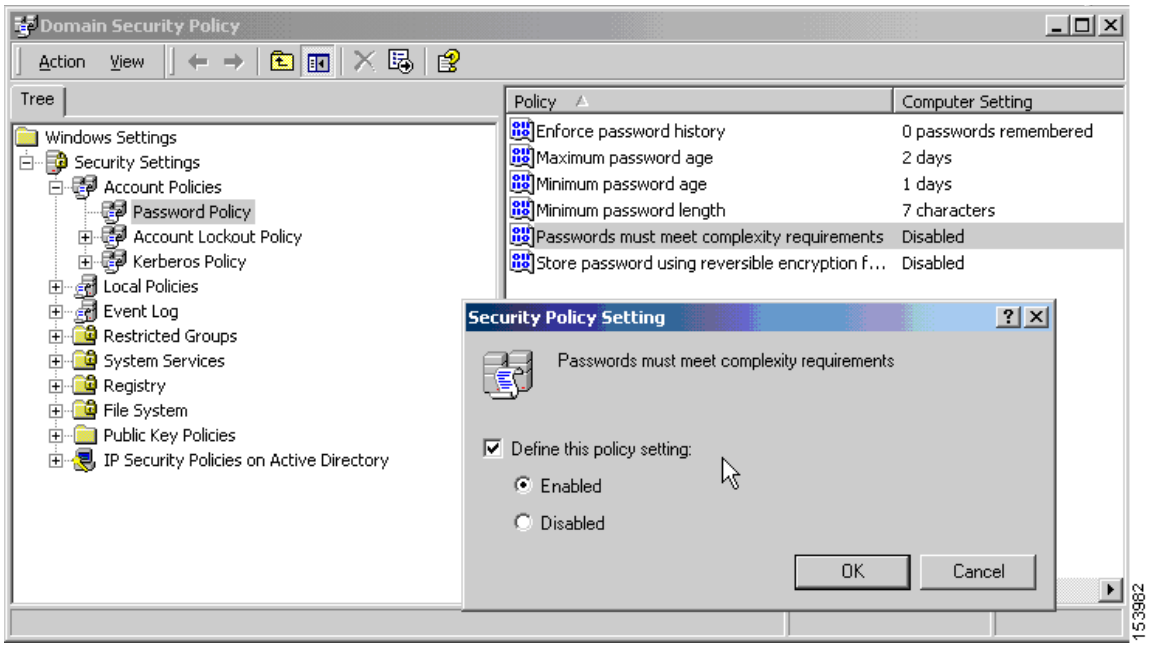
To enforce complex passwords—for example, to require that a password contain upper- and lowercase letters, numbers, and special characters—specify the **password-management** command in tunnel-group general-attributes configuration mode on the security appliance and do the following steps under Active Directory:

Step 1 Select Start > Programs > Administrative Tools > Domain Security Policy. Select Windows Settings > Security Settings > Account Policies > Password Policy.

Step 2 Double-click Password must meet complexity requirements to open the Security Policy Setting dialog box.

Step 3 Check the Define this policy setting check box and select Enable.

Figure 32-6 Active Directory—Enforce Password Complexity



Enforcing password complexity takes effect only when the user changes passwords; for example, when you have configured Enforce password change at next login or Password expires in *n* days. At login, the user receives a prompt to enter a new password, and the system will accept only a complex password.

Configuring the Connection Profile for RADIUS/SDI Message Support for the AnyConnect Client

This section describes procedures to ensure that the AnyConnect VPN client using RSA SecureID Software tokens can properly respond to user prompts delivered to the client through a RADIUS server proxying to an SDI server(s). This section contains the following topics:

- AnyConnect Client and RADIUS/SDI Server Interaction
- Configuring the Security Appliance to Support RADIUS/SDI Messages

AnyConnect Client and RADIUS/SDI Server Interaction

When a remote user connects to the security appliance with the AnyConnect VPN client and attempts to authenticate using an RSA SecurID token, the security appliance communicates with the RADIUS server, which in turn, communicates with the SDI server about the authentication.

During authentication, the RADIUS server presents access challenge messages to the security appliance. Within these challenge messages are reply messages containing text from the SDI server. The message text is different when the security appliance is communicating directly with an SDI server than when communicating through the RADIUS proxy. Therefore, in order to appear as a native SDI server to the AnyConnect client, the security appliance must interpret the messages from the RADIUS server.

Also, because the SDI messages are configurable on the SDI server, the message text on the security appliance must match (in whole or in part) the message text on the SDI server. Otherwise, the prompts displayed to the remote client user may not be appropriate for the action required during authentication. The AnyConnect client may fail to respond and authentication may fail.

The following section describes how to configure the security appliance to ensure successful authentication between the client and the SDI server:

Configuring the Security Appliance to Support RADIUS/SDI Messages

The following section describes the steps to configure the security appliance to interpret SDI-specific RADIUS reply messages and prompt the AnyConnect user for the appropriate action:

Step 1 Configure a connection profile (tunnel group) to forward RADIUS reply messages in a manner that simulates direct communication with an SDI server using the **proxy-auth sdi** command from tunnel-group webvpn configuration mode. Users authenticating to the SDI server must connect over this connection profile.

For example:

```
hostname(config)# tunnel-group sales webvpn attributes
hostname(tunnel-group-webvpn)# proxy-auth sdi
```

Step 2 Configure the RADIUS reply message text on the security appliance to match (in whole or in part) the message text sent by the RADIUS server with the **proxy-auth_map sdi** command from tunnel-group webvpn configuration mode.

The default message text used by the security appliance is the default message text used by Cisco Secure Access Control Server (ACS). If you are using Cisco Secure ACS, and it is using the default message text, you do not need to configure the message text on the security appliance. Otherwise, use the **proxy-auth_map sdi** command to ensure the message text matches.

Table 32-3 shows the message code, the default RADIUS reply message text, and the function of each message. Because the security appliance searches for strings in the order that they appear in the table, you must ensure that the string you use for the message text is not a subset of another string.

For example, "new PIN" is a subset of the default message text for both new-pin-sup and next-ccode-and-reauth. If you configure new-pin-sup as "new PIN", when the security appliance receives "new PIN with the next card code" from the RADIUS server, it will match the text to the new-pin-sup code instead of the next-ccode-and-reauth code.

Table 32-3 SDI Op-codes, Default Message Text, and Message Function

Message Code	Default RADIUS Reply Message Text	Function
next-code	Enter Next PASSCODE	Indicates the user must enter the NEXT tokencode without the PIN.
new-pin-sup	Please remember your new PIN	Indicates the new system PIN has been supplied and displays that PIN for the user.
new-pin-meth	Do you want to enter your own pin	Requests from the user which new PIN method to use to create a new PIN.
new-pin-req	Enter your new Alpha-Numerical PIN	Indicates a user-generated PIN and requests that the user enter the PIN.

Message Code	Default RADIUS Reply Message Text	Function
new-pin-reenter	Reenter PIN:	Used internally by the security appliance for user-supplied PIN confirmation. The client confirms the PIN without prompting the user.
new-pin-sys-ok	New PIN Accepted	Indicates the user-supplied PIN was accepted.
next-ccode-and-reauth	new PIN with the next card code	Follows a PIN operation and indicates the user must wait for the next tokencode and to enter both the new PIN and next tokencode to authenticate.
ready-for-sys-pin	ACCEPT A SYSTEM GENERATED PIN	Used internally by the security appliance to indicate the user is ready for the system-generated PIN.

The following example enters aaa-server-host mode and changes the text for the RADIUS reply message new-pin-sup:

```
hostname(config)# aaa-server radius_sales host 10.10.10.1
hostname(config-aaa-server-host)# proxy-auth_map sdi new-pin-sup "This is your new PIN"
```

Group Policies

This section describes group policies and how to configure them. It includes the following sections:

- Default Group Policy, page 32-37
- Configuring Group Policies, page 32-38

A group policy is a set of user-oriented attribute/value pairs for IPSec connections that are stored either internally (locally) on the device or externally on a RADIUS server. The connection profile uses a group policy that sets terms for user connections after the tunnel is established. Group policies let you apply whole sets of attributes to a user or a group of users, rather than having to specify each attribute individually for each user.

Enter the **group-policy** commands in global configuration mode to assign a group policy to users or to modify a group policy for specific users.

The security appliance includes a default group policy. In addition to the default group policy, which you can modify but not delete, you can create one or more group policies specific to your environment.

You can configure internal and external group policies. Internal groups are configured on the security appliance's internal database. External groups are configured on an external authentication server, such as RADIUS. Group policies include the following attributes:

- Identity
- Server definitions
- Client firewall settings
- Tunneling protocols
- IPSec settings
- Hardware client settings
- Filters
- Client configuration settings
- Connection settings

Default Group Policy

The security appliance supplies a default group policy. You can modify this default group policy, but you cannot delete it. A default group policy, named DfltGrpPolicy, always exists on the security appliance, but this default group policy does not take effect unless you configure the security appliance to use it. When you configure other group policies, any attribute that you do not explicitly specify takes its value from the default group policy. To view the default group policy, enter the following command:

```
hostname(config)# show running-config all group-policy DfltGrpPolicy
hostname(config)#
```

To configure the default group policy, enter the following command:

```
hostname(config)# group-policy DfltGrpPolicy internal
hostname(config)#
```

Note The default group policy is always internal. Despite the fact that the command syntax is `hostname(config)# group-policy DfltGrpPolicy {internal | external}`, you cannot change its type to external.

To change any of the attributes of the default group policy, use the **group-policy attributes** command to enter attributes mode, then specify the commands to change whatever attributes that you want to modify:

```
hostname(config)# group-policy DfltGrpPolicy attributes
```

Note The attributes mode applies only to internal group policies.

The default group policy, DfltGrpPolicy, that the security appliance provides is as follows:

```
group-policy DfltGrpPolicy internal
group-policy DfltGrpPolicy attributes
 banner none
 wins-server none
 dns-server none
 dhcp-network-scope none
 vpn-access-hours none
 vpn-simultaneous-logins 2000
 vpn-idle-timeout none
 vpn-session-timeout none
 vpn-filter none
 vpn-tunnel-protocol IPSec webvpn
 password-storage enable
 ip-comp disable
 re-xauth disable
 group-lock none
 pfs disable
 ipsec-udp disable
 ipsec-udp-port 10000
 split-tunnel-policy tunnelall
 split-tunnel-network-list none
 default-domain none
 split-dns none
 intercept-dhcp 255.255.255.255 disable
 secure-unit-authentication disable
```

```
 user-authentication disable
 user-authentication-idle-timeout 30
 ip-phone-bypass disable
 leap-bypass disable
 nem disable
 backup-servers keep-client-config
 msie-proxy server none
 msie-proxy method no-modify
 msie-proxy except-list none
 msie-proxy local-bypass disable
 nac disable
 nac-sq-period 300
 nac-reval-period 36000
 nac-default-acl none
 address-pools value vpn_users
 client-firewall none
 client-access-rule none
 webvpn
  html-content-filter none
  homepage none
  keep-alive-ignore 4
  http-comp gzip
  filter none
  url-list value MyURLs
  customization value DfltCustomization
  port-forward none
  port-forward-name value Application Access
  sso-server none
  deny-message value Login was successful, but because certain criteria have not been met
or due to some specific group policy, you do not have permission to use any of the VPN
features. Contact your IT administrator for more information
  svc none
  svc keep-installer none
  svc keepalive none
  svc rekey time none
  svc rekey method none
  svc dpd-interval client none
  svc dpd-interval gateway none
  svc compression deflate
 no vpn-nac-exempt
hostname(config-group-policy)#
```

You can modify the default group policy, and you can also create one or more group policies specific to your environment.

Configuring Group Policies

A group policy can apply to any kind of tunnel. In each case, if you do not explicitly define a parameter, the group takes the value from the default group policy. To configure a group policy, follow the steps in the subsequent sections.

Configuring an External Group Policy

External group policies take their attribute values from the external server that you specify. For an external group policy, you must identify the AAA server group that the security appliance can query for attributes and specify the password to use when retrieving attributes from the external AAA server

group. If you are using an external authentication server, and if your external group-policy attributes exist in the same RADIUS server as the users that you plan to authenticate, you have to make sure that there is no name duplication between them.

Note External group names on the security appliance refer to user names on the RADIUS server. In other words, if you configure external group X on the security appliance, the RADIUS server sees the query as an authentication request for user X. So external groups are really just user accounts on the RADIUS server that have special meaning to the security appliance. If your external group attributes exist in the same RADIUS server as the users that you plan to authenticate, there must be no name duplication between them.

The security appliance supports user authorization on an external LDAP or RADIUS server. Before you configure the security appliance to use an external server, you must configure the server with the correct security appliance authorization attributes and, from a subset of these attributes, assign specific permissions to individual users. Follow the instructions in Appendix D, "Configuring an External Server for Authorization and Authentication" to configure your external server.

To configure an external group policy, do the following steps specify a name and type for the group policy, along with the server-group name and a password:

```
hostname(config)# group-policy group_policy_name type server-group server_group_name
password server_password
hostname(config)#
```

Note For an external group policy, RADIUS is the only supported AAA server type.

For example, the following command creates an external group policy named ExtGroup that gets its attributes from an external RADIUS server named ExtRAD and specifies that the password to use when retrieving the attributes is newpassword:

```
hostname(config)# group-policy ExtGroup external server-group ExtRAD password newpassword
hostname(config)#
```

Note You can configure several vendor-specific attributes (VSAs), as described in Appendix D, "Configuring an External Server for Authorization and Authentication". If a RADIUS server is configured to return the Class attribute (#25), the security appliance uses that attribute to authenticate the Group Name. On the RADIUS server, the attribute must be formatted as: OU=*groupname*; where *groupname* is identical to the Group Name configured on the security appliance—for example, OU=Finance.

Configuring an Internal Group Policy

To configure an internal group policy, specify a name and type for the group policy:

```
hostname(config)# group-policy group_policy_name type
hostname(config)#
```

For example, the following command creates the internal group policy named GroupPolicy1:

```
hostname(config)# group-policy GroupPolicy1 internal
hostname(config)#
```

The default type is **internal**.

You can initialize the attributes of an internal group policy to the values of a preexisting group policy by appending the keyword **from** and specifying the name of the existing policy:

```
hostname(config)# group-policy group_policy_name internal from group_policy_name
hostname(config-group-policy)#
hostname(config-group-policy)#
```

Configuring Group Policy Attributes

For internal group policies, you can specify particular attribute values. To begin, enter group-policy attributes mode, by entering the **group-policy attributes** command in global configuration mode.

```
hostname(config)# group-policy name attributes
hostname(config-group-policy)#
```

The prompt changes to indicate the mode change. The group-policy-attributes mode lets you configure attribute-value pairs for a specified group policy. In group-policy-attributes mode, explicitly configure the attribute-value pairs that you do not want to inherit from the default group. The commands to do this are described in the following sections.

Configuring WINS and DNS Servers

You can specify primary and secondary WINS servers and DNS servers. The default value in each case is none. To specify these servers, do the following steps:

Step 1 Specify the primary and secondary WINS servers:

```
hostname(config-group-policy)# wins-server value {ip_address [ip_address] | none}
hostname(config-group-policy)#
```

The first IP address specified is that of the primary WINS server. The second (optional) IP address is that of the secondary WINS server. Specifying the **none** keyword instead of an IP address sets WINS servers to a null value, which allows no WINS servers and prevents inheriting a value from a default or specified group policy.

Every time that you enter the **wins-server** command, you overwrite the existing setting. For example, if you configure WINS server x.x.x.x and then configure WINS server y.y.y.y, the second command overwrites the first, and y.y.y.y becomes the sole WINS server. The same is true for multiple servers. To add a WINS server rather than overwrite previously configured servers, include the IP addresses of all WINS servers when you enter this command.

The following example shows how to configure WINS servers with the IP addresses 10.10.10.15 and 10.10.10.30 for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# wins-server value 10.10.10.15 10.10.10.30
hostname(config-group-policy)#
```

Step 2 Specify the primary and secondary DNS servers:

```
hostname(config-group-policy)# dns-server value {ip_address [ip_address] | none}
hostname(config-group-policy)#
```

The first IP address specified is that of the primary DNS server. The second (optional) IP address is that of the secondary DNS server. Specifying the **none** keyword instead of an IP address sets DNS servers to a null value, which allows no DNS servers and prevents inheriting a value from a default or specified group policy.

Every time that you enter the **dns-server** command you overwrite the existing setting. For example, if you configure DNS server x.x.x.x and then configure DNS server y.y.y.y, the second command overwrites the first, and y.y.y.y becomes the sole DNS server. The same is true for multiple servers. To add a DNS server rather than overwrite previously configured servers, include the IP addresses of all DNS servers when you enter this command.

The following example shows how to configure DNS servers with the IP addresses 10.10.10.15, and 10.10.10.30 for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# dns-server value 10.10.10.15 10.10.10.30
hostname(config-group-policy)#
```

Step 3 Configure the DHCP network scope:

```
hostname(config-group-policy)# dhcp-network-scope {ip_address | none}
hostname(config-group-policy)#
```

DHCP scope specifies the range of IP addresses (that is, a subnetwork) that the security appliance DHCP server should use to assign addresses to users of this group policy.

The following example shows how to set an IP subnetwork of 10.10.85.0 (specifying the address range of 10.10.85.0 through 10.10.85.255) for the group policy named First Group:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# dhcp-network-scope 10.10.85.0
```

Configuring VPN-Specific Attributes

Follow the steps in this section to set the VPN attribute values. The VPN attributes control the access hours, the number of simultaneous logins allowed, the timeouts, the egress VLAN or ACL to apply to VPN sessions, and the tunnel protocol:

Step 1 Set the VPN access hours. To do this, you associate a group policy with a configured time-range policy, using the **vpn-access-hours** command in group-policy configuration mode.

```
hostname(config-group-policy)# vpn-access-hours value {time-range | none}
```

A group policy can inherit a time-range value from a default or specified group policy. To prevent this inheritance, enter the **none** keyword instead of the name of a time-range in this command. This keyword sets VPN access hours to a null value, which allows no time-range policy.

The time-range variable is the name of a set of access hours defined in global configuration mode using the **time-range** command. The following example shows how to associate the group policy named FirstGroup with a time-range policy called 824:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# vpn-access-hours value 824
```

Step 2 Specify the number of simultaneous logins allowed for any user, using the **vpn-simultaneous-logins** command in group-policy configuration mode.

```
hostname(config-group-policy)# vpn-simultaneous-logins integer
```

The default value is 3. The range is an integer in the range 0 through 2147483647. A group policy can inherit this value from another group policy. Enter 0 to disable login and prevent user access. The following example shows how to allow a maximum of 4 simultaneous logins for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# vpn-simultaneous-logins 4
hostname(config-group-policy)#
```

Note While the maximum limit for the number of simultaneous logins is very large, allowing several simultaneous logins could compromise security and affect performance.

Stale AnyConnect, IPSec Client, or Clientless sessions (sessions that are terminated abnormally) might remain in the session database, even though a "new" session has been established with the same username.

If the value of vpn-simultaneous-logins is 1, and the same user logs in again after an abnormal termination, then the stale session is removed from the database and the new session is established. If, however, the existing session is still an active connection and the same user logs in again, perhaps from another PC, the first session is logged off and removed from the database, and the new session is established.

If the number of simultaneous logins is a value greater than 1, then, when you have reached that maximum number and try to log in again, the session with the longest idle time is logged off. If all current sessions have been idle an equally long time, then the oldest session is logged off. This action frees up a session and allows the new login.

Step 3 Configure the user timeout period by entering the **vpn-idle-timeout** command in group-policy configuration mode or in username configuration mode:

```
hostname(config-group-policy)# vpn-idle-timeout {minutes | none}
hostname(config-group-policy)#
```

The minimum time is 1 minute, and the maximum time is 35791394 minutes. The default is 30 minutes. If there is no communication activity on the connection in this period, the security appliance terminates the connection.

A group policy can inherit this value from another group policy. To prevent inheriting a value, enter the **none** keyword instead of specifying a number of minutes with this command. The none keyword also permits an unlimited idle timeout period. It sets the idle timeout to a null value, thereby disallowing an idle timeout.

The following example shows how to set a VPN idle timeout of 15 minutes for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# vpn-idle-timeout 15
hostname(config-group-policy)#
```

Step 4 Configure a maximum amount of time for VPN connections, using the **vpn-session-timeout** command in group-policy configuration mode or in username configuration mode.

```
hostname(config-group-policy)# vpn-session-timeout {minutes | none}
hostname(config-group-policy)#
```

The minimum time is 1 minute, and the maximum time is 35791394 minutes. There is no default value. At the end of this period of time, the security appliance terminates the connection.

A group policy can inherit this value from another group policy. To prevent inheriting a value, enter the **none** keyword instead of specifying a number of minutes with this command. Specifying the **none** keyword permits an unlimited session timeout period and sets session timeout with a null value, which disallows a session timeout.

The following example shows how to set a VPN session timeout of 180 minutes for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# vpn-session-timeout 180
hostname(config-group-policy)#
```

Step 5 Choose one of the following options to specify an egress VLAN (also called "VLAN mapping") for remote access or specify an ACL to filter the traffic:

- Enter the following command in group-policy configuration mode to specify the egress VLAN for remote access VPN sessions assigned to this group policy or to a group policy that inherits this group policy:

  ```
  hostname(config-group-policy)# [no] vlan {vlan_id |none}
  ```

  no vlan removes the *vlan_id* from the group policy. The group policy inherits the vlan value from the default group policy.

  vlan none removes the *vlan_id* from the group policy and disables VLAN mapping for this group policy. The group policy does not inherit the vlan value from the default group policy.

  vlan_id in the command **vlan** *vlan_id* is the number of the VLAN, in decimal format, to assign to remote access VPN sessions that use this group policy. The VLAN must be configured on this security appliance per the instructions in "Configuring VLAN Subinterfaces and 802.1Q Trunking" procedure on page 6-7.

  none disables the assignment of a VLAN to the remote access VPN sessions that match this group policy.

  Note The egress VLAN feature works for HTTP connections, but not for FTP and CIFS.

- Specify the name of the ACL to apply to VPN session, using the **vpn-filter** command in group policy mode. (You can also configure this attribute in username mode, in which case the value configured under username supersedes the group-policy value.)

  ```
  hostname(config-group-policy)# vpn-filter {value ACL name | none}
  hostname(config-group-policy)#
  ```

  You configure ACLs to permit or deny various types of traffic for this group policy. You then enter the **vpn-filter** command to apply those ACLs.

  To remove the ACL, including a null value created by entering the **vpn-filter none** command, enter the **no** form of this command. The **no** option allows inheritance of a value from another group policy.

  A group policy can inherit this value from another group policy. To prevent inheriting a value, enter the **none** keyword instead of specifying an ACL name. The **none** keyword indicates that there is no access list and sets a null value, thereby disallowing an access list.

  The following example shows how to set a filter that invokes an access list named acl_vpn for the group policy named FirstGroup:

  ```
  hostname(config)# group-policy FirstGroup attributes
  hostname(config-group-policy)# vpn-filter acl_vpn
  hostname(config-group-policy)#
  ```

Step 6 Specify the VPN tunnel type for this group policy.

```
hostname(config-group-policy)# vpn-tunnel-protocol {webvpn | IPSec | l2tp-ipsec}
hostname(config-group-policy)#
```

The default is IPSec. To remove the attribute from the running configuration, enter the **no** form of this command.

```
hostname(config-group-policy)# no vpn-tunnel-protocol [webvpn | IPSec | l2tp-ipsec]
hostname(config-group-policy)#
```

The parameter values for this command follow:

- **IPSec**—Negotiates an IPSec tunnel between two peers (a remote access client or another secure gateway). Creates security associations that govern authentication, encryption, encapsulation, and key management.
- **webvpn**—Provides VPN services to remote users via an HTTPS-enabled web browser, and does not require a client.
- **l2tp-ipsec**—Negotiates an IPSec tunnel for an L2TP connection

Enter this command to configure one or more tunneling modes. You must configure at least one tunneling mode for users to connect over a VPN tunnel.

The following example shows how to configure the IPSec tunneling mode for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# vpn-tunnel-protocol IPSec
hostname(config-group-policy)#
```

Configuring Security Attributes

The attributes in this section specify certain security settings for the group:

Step 1 Specify whether to let users store their login passwords on the client system, using the **password-storage** command with the **enable** keyword in group-policy configuration mode. To disable password storage, use the **password-storage** command with the **disable** keyword.

```
hostname(config-group-policy)# password-storage {enable | disable}
hostname(config-group-policy)#
```

For security reasons, password storage is disabled by default. Enable password storage only on systems that you know to be in secure sites.

To remove the password-storage attribute from the running configuration, enter the **no** form of this command:

```
hostname(config-group-policy)# no password-storage
hostname(config-group-policy)#
```

Specifying the **no** form enables inheritance of a value for password-storage from another group policy.

This command does not apply to interactive hardware client authentication or individual user authentication for hardware clients.

The following example shows how to enable password storage for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# password-storage enable
hostname(config-group-policy)#
```

Step 2 Specify whether to enable IP compression, which is disabled by default.

```
hostname(config-group-policy)# ip-comp {enable | disable}
```

```
hostname(config-group-policy)#
```

To enable LZS IP compression, enter the **ip-comp** command with the **enable** keyword in group-policy configuration mode. To disable IP compression, enter the **ip-comp** command with the **disable** keyword.

To remove the **ip-comp** attribute from the running configuration, enter the **no** form of this command. This enables inheritance of a value from another group policy.

```
hostname(config-group-policy)# no ip-comp
hostname(config-group-policy)#
```

Enabling data compression might speed up data transmission rates for remote dial-in users connecting with modems.

Caution Data compression increases the memory requirement and CPU usage for each user session and consequently decreases the overall throughput of the security appliance. For this reason, we recommend that you enable data compression only for remote users connecting with a modem. Design a group policy specific to modem users, and enable compression only for them.

Step 3 Specify whether to require that users reauthenticate on IKE rekey by using the **re-xauth** command with the **enable** keyword in group-policy configuration mode. If you enable reauthentication on IKE rekey, the security appliance prompts the user to enter a username and password during initial Phase 1 IKE negotiation and also prompts for user authentication whenever an IKE rekey occurs. Reauthentication provides additional security.

If the configured rekey interval is very short, users might find the repeated authorization requests inconvenient. To avoid repeated authorization requests, disable reauthentication. To check the configured rekey interval, in monitoring mode, enter the **show crypto ipsec sa** command to view the security association lifetime in seconds and lifetime in kilobytes of data. To disable user reauthentication on IKE rekey, enter the **disable** keyword. Reauthentication on IKE rekey is disabled by default.

```
hostname(config-group-policy)# re-xauth {enable | disable}
hostname(config-group-policy)#
```

To enable inheritance of a value for reauthentication on IKE rekey from another group policy, remove the re-xauth attribute from the running configuration by entering the **no** form of this command.

```
hostname(config-group-policy)# no re-xauth
hostname(config-group-policy)#
```

Note Reauthentication fails if there is no user at the other end of the connection.

Step 4 Specify whether to restrict remote users to access only through the connection profile, using the **group-lock** command in group-policy configuration mode.

```
hostname(config-group-policy)# group-lock {value tunnel-grp-name | none}
hostname(config-group-policy)# no group-lock
hostname(config-group-policy)#
```

The *tunnel-grp-name* variable specifies the name of an existing connection profile that the security appliance requires for the user to connect. Group-lock restricts users by checking if the group configured in the VPN client is the same as the connection profile to which the user is assigned. If it is not, the security appliance prevents the user from connecting. If you do not configure group-lock, the security appliance authenticates users without regard to the assigned group. Group locking is disabled by default.

To remove the **group-lock** attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a value from another group policy.

To disable group-lock, enter the **group-lock** command with the **none** keyword. The none keyword sets group-lock to a null value, thereby allowing no group-lock restriction. It also prevents inheriting a group-lock value from a default or specified group policy

Step 5 Specify whether to enable perfect forward secrecy. In IPSec negotiations, perfect forward secrecy ensures that each new cryptographic key is unrelated to any previous key. A group policy can inherit a value for perfect forward secrecy from another group policy. Perfect forward secrecy is disabled by default. To enable perfect forward secrecy, use the **pfs** command with the **enable** keyword in group-policy configuration mode.

```
hostname(config-group-policy)# pfs {enable | disable}
hostname(config-group-policy)#
```

To disable perfect forward secrecy, enter the **pfs** command with the **disable** keyword.

To remove the perfect forward secrecy attribute from the running configuration and prevent inheriting a value, enter the **no** form of this command.

```
hostname(config-group-policy)# no pfs
hostname(config-group-policy)#
```

Configuring the Banner Message

Specify the banner, or welcome message, if any, that you want to display. The default is no banner. The message that you specify is displayed on remote clients when they connect. To specify a banner, enter the **banner** command in group-policy configuration mode. The banner text can be up to 510 characters long. Enter the "\n" sequence to insert a carriage return.

Note A carriage-return/line-feed included in the banner counts as two characters.

To delete a banner, enter the **no** form of this command. Be aware that using the **no** version of the command deletes all banners for the group policy.

A group policy can inherit this value from another group policy. To prevent inheriting a value, enter the **none** keyword instead of specifying a value for the banner string, as follows:

```
hostname(config-group-policy)# banner {value banner_string | none}
```

The following example shows how to create a banner for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# banner value Welcome to Cisco Systems 7.0.
```

Configuring IPSec-UDP Attributes

IPSec over UDP, sometimes called IPSec through NAT, lets a Cisco VPN client or hardware client connect via UDP to a security appliance that is running NAT. It is disabled by default. IPSec over UDP is proprietary; it applies only to remote-access connections, and it requires mode configuration. The security appliance exchanges configuration parameters with the client while negotiating SAs. Using IPSec over UDP may slightly degrade system performance.

To enable IPSec over UDP, configure the **ipsec-udp** command with the **enable** keyword in group-policy configuration mode, as follows:

```
hostname(config-group-policy)# ipsec-udp {enable | disable}
```

```
hostname(config-group-policy)# no ipsec-udp
```

To use IPSec over UDP, you must also configure the **ipsec-udp-port** command, as described below.

To disable IPSec over UDP, enter the **disable** keyword. To remove the IPSec over UDP attribute from the running configuration, enter the **no** form of this command. This enables inheritance of a value for IPSec over UDP from another group policy.

The Cisco VPN client must also be configured to use IPSec over UDP (it is configured to use it by default). The VPN 3002 requires no configuration to use IPSec over UDP.

The following example shows how to set IPSec over UDP for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# ipsec-udp enable
```

If you enabled IPSec over UDP, you must also configure the **ipsec-udp-port** command in group-policy configuration mode. This command sets a UDP port number for IPSec over UDP. In IPSec negotiations, the security appliance listens on the configured port and forwards UDP traffic for that port even if other filter rules drop UDP traffic. The port numbers can range from 4001 through 49151. The default port value is 10000.

To disable the UDP port, enter the **no** form of this command. This enables inheritance of a value for the IPSec over UDP port from another group policy.

```
hostname(config-group-policy)# ipsec-udp-port port
```

The following example shows how to set an IPSec UDP port to port 4025 for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# ipsec-udp-port 4025
```

Configuring Split-Tunneling Attributes

Split tunneling lets a remote-access IPSec client conditionally direct packets over an IPSec tunnel in encrypted form or to a network interface in clear text form. With split tunneling enabled, packets not bound for destinations on the other side of the IPSec tunnel do not have to be encrypted, sent across the tunnel, decrypted, and then routed to a final destination. This command applies this split tunneling policy to a specific network.

Setting the Split-Tunneling Policy

Set the rules for tunneling traffic by specifying the split-tunneling policy:

```
hostname(config-group-policy)# split-tunnel-policy {tunnelall | tunnelspecified |
excludespecified}
hostname(config-group-policy)# no split-tunnel-policy
```

The default is to tunnel all traffic. To set a split tunneling policy, enter the **split-tunnel-policy** command in group-policy configuration mode. To remove the **split-tunnel-policy** attribute from the running configuration, enter the **no** form of this command. This enables inheritance of a value for split tunneling from another group policy.

The **excludespecified** keyword defines a list of networks to which traffic goes in the clear. This feature is useful for remote users who want to access devices on their local network, such as printers, while they are connected to the corporate network through a tunnel. This option applies only to the Cisco VPN client.

The **tunnelall** keyword specifies that no traffic goes in the clear or to any other destination than the security appliance. This, in effect, disables split tunneling. Remote users reach Internet networks through the corporate network and do not have access to local networks. This is the default option.

The **tunnelspecified** keyword tunnels all traffic from or to the specified networks. This option enables split tunneling. It lets you create a network list of addresses to tunnel. Data to all other addresses travels in the clear and is routed by the remote user's Internet service provider.

Note Split tunneling is primarily a traffic management feature, not a security feature. For optimum security, we recommend that you do not enable split tunneling.

The following example shows how to set a split tunneling policy of tunneling only specified networks for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# split-tunnel-policy tunnelspecified
```

Creating a Network List for Split-Tunneling

Create a network list for split tunneling using the **split-tunnel-network-list** command in group-policy configuration mode.

```
hostname(config-group-policy)# split-tunnel-network-list {value access-list_name | none}
hostname(config-group-policy)# no split-tunnel-network-list value [access-list_name]
```

Split tunneling network lists distinguish networks that require traffic to travel across the tunnel from those that do not require tunneling. The security appliance makes split tunneling decisions on the basis of a network list, which is an ACL that consists of a list of addresses on the private network. Only standard-type ACLs are allowed.

The **value** *access-list name* parameter identifies an access list that enumerates the networks to tunnel or not tunnel.

The **none** keyword indicates that there is no network list for split tunneling; the security appliance tunnels all traffic. Specifying the **none** keyword sets a split tunneling network list with a null value, thereby disallowing split tunneling. It also prevents inheriting a default split tunneling network list from a default or specified group policy.

To delete a network list, enter the **no** form of this command. To delete all split tunneling network lists, enter the **no split-tunnel-network-list** command without arguments. This command deletes all configured network lists, including a null list if you created one by entering the **none** keyword.

When there are no split tunneling network lists, users inherit any network lists that exist in the default or specified group policy. To prevent users from inheriting such network lists, enter the **split-tunnel-network-list none** command.

The following example shows how to set a network list called FirstList for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# split-tunnel-network-list FirstList
```

Configuring Domain Attributes for Tunneling

You can specify a default domain name for tunneled packets or a list of domains to be resolved through the split tunnel. The following sections describe how to set these domains.

Defining a Default Domain Name for Tunneled Packets

The security appliance passes the default domain name to the IPSec client to append to DNS queries that omit the domain field. When there are no default domain names, users inherit the default domain name in the default group policy. To specify the default domain name for users of the group policy, enter the **default-domain** command in group-policy configuration mode. To delete a domain name, enter the **no** form of this command.

```
hostname(config-group-policy)# default-domain {value domain-name | none}
hostname(config-group-policy)# no default-domain [domain-name]
```

The **value** *domain-name* parameter identifies the default domain name for the group. To specify that there is no default domain name, enter the **none** keyword. This command sets a default domain name with a null value, which disallows a default domain name and prevents inheriting a default domain name from a default or specified group policy.

To delete all default domain names, enter the **no default-domain** command without arguments. This command deletes all configured default domain names, including a null list if you created one by entering the **default-domain** command with the **none** keyword. The **no** form allows inheriting a domain name.

The following example shows how to set a default domain name of FirstDomain for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# default-domain value FirstDomain
```

Defining a List of Domains for Split Tunneling

Enter a list of domains to be resolved through the split tunnel. Enter the **split-dns** command in group-policy configuration mode. To delete a list, enter the **no** form of this command.

Note The AnyConnect client does not support split DNS.

When there are no split tunneling domain lists, users inherit any that exist in the default group policy. To prevent users from inheriting such split tunneling domain lists, enter the **split-dns** command with the **none** keyword.

To delete all split tunneling domain lists, enter the **no split-dns** command without arguments. This deletes all configured split tunneling domain lists, including a null list created by issuing the **split-dns** command with the **none** keyword.

The parameter **value** *domain-name* provides a domain name that the security appliance resolves through the split tunnel. The **none** keyword indicates that there is no split DNS list. It also sets a split DNS list with a null value, thereby disallowing a split DNS list, and prevents inheriting a split DNS list from a default or specified group policy. The syntax of the command is as follows:

```
hostname(config-group-policy)# split-dns {value domain-name1 [domain-name2...
domain-nameN] | none}
hostname(config-group-policy)# no split-dns [domain-name domain-name2 domain-nameN]
```

Enter a single space to separate each entry in the list of domains. There is no limit on the number of entries, but the entire string can be no longer than 255 characters. You can use only alphanumeric characters, hyphens (-), and periods (.). If the default domain name is to be resolved through the tunnel, you must explicitly include that name in this list.

The following example shows how to configure the domains Domain1, Domain2, Domain3, and Domain4 to be resolved through split tunneling for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# split-dns value Domain1 Domain2 Domain3 Domain4
```

Configuring DHCP Intercept

A Microsoft XP anomaly results in the corruption of domain names if split tunnel options exceed 255 bytes. To avoid this problem, the security appliance limits the number of routes it sends to 27 to 40 routes, with the number of routes dependent on the classes of the routes.

DHCP Intercept lets Microsoft Windows XP clients use split-tunneling with the security appliance. The security appliance replies directly to the Microsoft Windows XP client DHCP Inform message, providing that client with the subnet mask, domain name, and classless static routes for the tunnel IP address. For Windows clients prior to Windows XP, DHCP Intercept provides the domain name and subnet mask. This is useful in environments in which using a DHCP server is not advantageous.

The **intercept-dhcp** command enables or disables DHCP intercept. The syntax of this command is as follows:

[no] intercept-dhcp

```
hostname(config-group-policy)# intercept-dhcp netmask {enable | disable}
hostname(config-group-policy)#
```

The *netmask* variable provides the subnet mask for the tunnel IP address. The **no** version of the command removes the DHCP intercept from the configuration.

The following example shows how to set DHCP Intercepts for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# intercept-dhcp enable
```

Configuring Attributes for VPN Hardware Clients

The commands in this section enable or disable secure unit authentication and user authentication, and set a user authentication timeout value for VPN hardware clients. They also let you allow Cisco IP phones and LEAP packets to bypass individual user authentication and allow hardware clients using Network Extension Mode to connect.

Configuring Secure Unit Authentication

Secure unit authentication provides additional security by requiring VPN hardware clients to authenticate with a username and password each time that the client initiates a tunnel. With this feature enabled, the hardware client does not have a saved username and password. Secure unit authentication is disabled by default.

Note With this feature enabled, to bring up a VPN tunnel, a user must be present to enter the username and password.

Secure unit authentication requires that you have an authentication server group configured for the connection profile the hardware client(s) use. If you require secure unit authentication on the primary security appliance, be sure to configure it on any backup servers as well.

Specify whether to enable secure unit authentication by entering the **secure-unit-authentication** command with the **enable** keyword in group-policy configuration mode.

```
hostname(config-group-policy)# secure-unit-authentication {enable | disable}
hostname(config-group-policy)# no secure-unit-authentication
```

To disable secure unit authentication, enter the **disable** keyword. To remove the secure unit authentication attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a value for secure unit authentication from another group policy.

The following example shows how to enable secure unit authentication for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# secure-unit-authentication enable
```

Configuring User Authentication

User authentication is disabled by default. When enabled, user authentication requires that individual users behind a hardware client authenticate to gain access to the network across the tunnel. Individual users authenticate according to the order of authentication servers that you configure.

Specify whether to enable user authentication by entering the **user-authentication** command with the **enable** keyword in group-policy configuration mode.

```
hostname(config-group-policy)# user-authentication {enable | disable}
hostname(config-group-policy)# no user-authentication
```

To disable user authentication, enter the **disable** keyword. To remove the user authentication attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a value for user authentication from another group policy.

If you require user authentication on the primary security appliance, be sure to configure it on any backup servers as well.

The following example shows how to enable user authentication for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# user-authentication enable
```

Configuring an Idle Timeout

Set an idle timeout for individual users behind hardware clients by entering the **user-authentication-idle-timeout** command in group-policy configuration mode. If there is no communication activity by a user behind a hardware client in the idle timeout period, the security appliance terminates the client's access:

```
hostname(config-group-policy)# user-authentication-idle-timeout {minutes | none}
hostname(config-group-policy)# no user-authentication-idle-timeout
```

Note This timer terminates only the client's access through the VPN tunnel, not the VPN tunnel itself.

The idle timeout indicated in response to the **show uauth** command is always the idle timeout value of the user who authenticated the tunnel on the Cisco Easy VPN remote device.

The *minutes* parameter specifies the number of minutes in the idle timeout period. The minimum is 1 minute, the default is 30 minutes, and the maximum is 35791394 minutes.

To delete the idle timeout value, enter the **no** form of this command. This option allows inheritance of an idle timeout value from another group policy.

To prevent inheriting an idle timeout value, enter the **user-authentication-idle-timeout** command with the **none** keyword. This command sets the idle timeout with a null value, which disallows an idle timeout and prevents inheriting an user authentication idle timeout value from a default or specified group policy.

The following example shows how to set an idle timeout value of 45 minutes for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# user-authentication-idle-timeout 45
```

Configuring IP Phone Bypass

You can allow Cisco IP phones to bypass individual user authentication behind a hardware client. To enable IP Phone Bypass, enter the **ip-phone-bypass** command with the **enable** keyword in group-policy configuration mode. IP Phone Bypass lets IP phones behind hardware clients connect without undergoing user authentication processes. IP Phone Bypass is disabled by default. If enabled, secure unit authentication remains in effect.

To disable IP Phone Bypass, enter the **disable** keyword. To remove the IP phone Bypass attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a value for IP Phone Bypass from another group policy:

```
hostname(config-group-policy)# ip-phone-bypass {enable | disable}
hostname(config-group-policy)# no ip-phone-bypass
```

Configuring LEAP Bypass

When LEAP Bypass is enabled, LEAP packets from wireless devices behind a VPN 3002 hardware client travel across a VPN tunnel prior to user authentication. This action lets workstations using Cisco wireless access point devices establish LEAP authentication and then authenticate again per user authentication. LEAP Bypass is disabled by default.

To allow LEAP packets from Cisco wireless access points to bypass individual users authentication, enter the **leap-bypass** command with the **enable** keyword in group-policy configuration mode. To disable LEAP Bypass, enter the **disable** keyword. To remove the LEAP Bypass attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a value for LEAP Bypass from another group policy:

```
hostname(config-group-policy)# leap-bypass {enable | disable}
hostname(config-group-policy)# no leap-bypass
```



Note IEEE 802.1X is a standard for authentication on wired and wireless networks. It provides wireless LANs with strong mutual authentication between clients and authentication servers, which can provide dynamic per-user, per session wireless encryption privacy (WEP) keys, removing administrative burdens and security issues that are present with static WEP keys.

Cisco Systems has developed an 802.1X wireless authentication type called Cisco LEAP. LEAP (Lightweight Extensible Authentication Protocol) implements mutual authentication between a wireless client on one side of a connection and a RADIUS server on the other side. The credentials used for authentication, including a password, are always encrypted before they are transmitted over the wireless medium.

Cisco LEAP authenticates wireless clients to RADIUS servers. It does not include RADIUS accounting services.

This feature does not work as intended if you enable interactive hardware client authentication.

Caution There might be security risks to your network in allowing any unauthenticated traffic to traverse the tunnel.

The following example shows how to set LEAP Bypass for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# leap-bypass enable
```

Enabling Network Extension Mode

Network extension mode lets hardware clients present a single, routable network to the remote private network over the VPN tunnel. IPSec encapsulates all traffic from the private network behind the hardware client to networks behind the security appliance. PAT does not apply. Therefore, devices behind the security appliance have direct access to devices on the private network behind the hardware client over the tunnel, and only over the tunnel, and vice versa. The hardware client must initiate the tunnel, but after the tunnel is up, either side can initiate data exchange.

Enable network extension mode for hardware clients by entering the **nem** command with the **enable** keyword in group-policy configuration mode:

```
hostname(config-group-policy)# nem {enable | disable}
hostname(config-group-policy)# no nem
```

To disable NEM, enter the **disable** keyword. To remove the NEM attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a value from another group policy.

The following example shows how to set NEM for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# nem enable
```

Configuring Backup Server Attributes

Configure backup servers if you plan on using them. IPSec backup servers let a VPN client connect to the central site when the primary security appliance is unavailable.When you configure backup servers, the security appliance pushes the server list to the client as the IPSec tunnel is established. Backup servers do not exist until you configure them, either on the client or on the primary security appliance.

Configure backup servers either on the client or on the primary security appliance. If you configure backup servers on the security appliance, it pushes the backup server policy to the clients in the group, replacing the backup server list on the client if one is configured.

Note If you are using hostnames, it is wise to have backup DNS and WINS servers on a separate network from that of the primary DNS and WINS servers. Otherwise, if clients behind a hardware client obtain DNS and WINS information from the hardware client via DHCP, and the connection to the primary server is

lost, and the backup servers have different DNS and WINS information, clients cannot be updated until the DHCP lease expires. In addition, if you use hostnames and the DNS server is unavailable, significant delays can occur.

To configure backup servers, enter the **backup-servers** command in group-policy configuration mode:

```
hostname(config-group-policy)# backup-servers {server1 server2... server10 |
clear-client-config | keep-client-config}
```

To remove a backup server, enter the **no** form of this command with the backup server specified. To remove the backup-servers attribute from the running configuration and enable inheritance of a value for backup-servers from another group policy, enter the **no** form of this command without arguments.

```
hostname(config-group-policy)# no backup-servers [server1 server2... server10 |
clear-client-config | keep-client-config]
```

The **clear-client-config** keyword specifies that the client uses no backup servers. The security appliance pushes a null server list.

The **keep-client-config** keyword specifies that the security appliance sends no backup server information to the client. The client uses its own backup server list, if configured. This is the default.

The *server1 server 2.... server10* parameter list is a space-delimited, priority-ordered list of servers for the VPN client to use when the primary security appliance is unavailable. This list identifies servers by IP address or hostname. The list can be 500 characters long, and it can contain up to10 entries.

The following example shows how to configure backup servers with IP addresses 10.10.10.1 and 192.168.10.14, for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# backup-servers 10.10.10.1 192.168.10.14
```

Configuring Microsoft Internet Explorer Client Parameters

The following commands configure the proxy server parameters for a Microsoft Internet Explorer client.

Step 1 Configure a Microsoft Internet Explorer browser proxy server and port for a client PC by entering the **msie-proxy server** command in group-policy configuration mode:

```
hostname(config-group-policy)# msie-proxy server {value server[:port] | none}
hostname(config-group-policy)#
```

The default value is **none**. To remove the attribute from the configuration, use the **no** form of the command.

```
hostname(config-group-policy)# no msie-proxy server
hostname(config-group-policy)#
```

The line containing the proxy server IP address or hostname and the port number must be less than 100 characters long.

The following example shows how to configure the IP address 192.168.10.1 as a Microsoft Internet Explorer proxy server, using port 880, for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# msie-proxy server value 192.168.21.1:880
hostname(config-group-policy)#
```

Step 2 Configure the Microsoft Internet Explorer browser proxy actions (“methods”) for a client PC by entering the **msie-proxy method** command in group-policy configuration mode.

```
hostname(config-group-policy)# msie-proxy method [auto-detect | no-modify | no-proxy |
use-server]
hostname(config-group-policy)#
```

The default value is **use-server**. To remove the attribute from the configuration, use the **no** form of the command.

```
hostname(config-group-policy)# no msie-proxy method [auto-detect |no-modify | no-proxy |
use-server]
hostname(config-group-policy)#
```

The available methods are as follows:

- **auto-detect**—Enables the use of automatic proxy server detection in Internet Explorer for the client PC.
- **no-modify**—Leaves the HTTP browser proxy server setting in Internet Explorer unchanged for this client PC.
- **no-proxy—**Disables the HTTP proxy setting in Internet Explorer for the client PC.
- **use-server—**Sets the HTTP proxy server setting in Internet Explorer to use the value configured in the **msie-proxy server** command.

The line containing the proxy server IP address or hostname and the port number must be less than 100 characters long.

The following example shows how to configure auto-detect as the Microsoft Internet Explorer proxy setting for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# msie-proxy method auto-detect
hostname(config-group-policy)#
```

The following example configures the Microsoft Internet Explorer proxy setting for the group policy named FirstGroup to use the server QAserver, port 1001 as the server for the client PC:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# msie-proxy server QAserver:port 1001
hostname(config-group-policy)# msie-proxy method use-server
hostname(config-group-policy)#
```

Step 3 Configure Microsoft Internet Explorer browser proxy exception list settings for a local bypass on the client PC by entering the **msie-proxy except-list** command in group-policy configuration mode. These addresses are not accessed by a proxy server. This list corresponds to the Exceptions box in the Proxy Settings dialog box in Internet Explorer.

```
hostname(config-group-policy)# msie-proxy except-list {value server[:port] | none}
hostname(config-group-policy)#
```

To remove the attribute from the configuration, use the **no** form of the command.

```
hostname(config-group-policy)# no msie-proxy except-list
hostname(config-group-policy)#
```

- **value** *server:port*—Specifies the IP address or name of an MSIE server and port that is applied for this client PC. The port number is optional.
- **none—**Indicates that there is no IP address/hostname or port and prevents inheriting an exception list.

By default, msie-proxy except-list is disabled.

The line containing the proxy server IP address or hostname and the port number must be less than 100 characters long.

The following example shows how to set a Microsoft Internet Explorer proxy exception list, consisting of the server at IP address 192.168.20.1, using port 880, for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# msie-proxy except-list value 192.168.20.1:880
hostname(config-group-policy)#
```

Step 4 Enable or disable Microsoft Internet Explorer browser proxy local-bypass settings for a client PC by entering the **msie-proxy local-bypass** command in group-policy configuration mode.

```
hostname(config-group-policy)# msie-proxy local-bypass {enable | disable}
hostname(config-group-policy)#
```

To remove the attribute from the configuration, use the **no** form of the command.

```
hostname(config-group-policy)# no msie-proxy local-bypass {enable | disable}
hostname(config-group-policy)#
```

By default, msie-proxy local-bypass is disabled.

The following example shows how to enable Microsoft Internet Explorer proxy local-bypass for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# msie-proxy local-bypass enable
hostname(config-group-policy)#
```

Configuring Network Admission Control Parameters

The group-policy NAC commands in this section all have default values. Unless you have a good reason for changing them, accept the default values for these parameters.

The security appliance uses Extensible Authentication Protocol (EAP) over UDP (EAPoUDP) messaging to validate the posture of remote hosts. Posture validation involves the checking of a remote host for compliancy with safety requirements before the assignment of a network access policy. An Access Control Server must be configured for Network Admission Control before you configure NAC on the security appliance.

The Access Control Server downloads the posture token, an informational text string configurable on the ACS, to the security appliance to aid in system monitoring, reporting, debugging, and logging. A typical posture token is Healthy, Checkup, Quarantine, Infected, or Unknown. Following posture validation or clientless authentication, the ACS downloads the access policy for the session to the security appliance.

The following parameters let you configure Network Admission Control settings for the default group policy or an alternative group policy.

Step 1 (*Optional*) Configure the status query timer period. The security appliance starts the status query timer after each successful posture validation and status query response. The expiration of this timer triggers a query for changes in the host posture, referred to as a status query. Enter the number of seconds in the range 30 through 1800. The default setting is 300.

To specify the interval between each successful posture validation in a Network Admission Control session and the next query for changes in the host posture, use the **nac-sq-period** command in group-policy configuration mode:

```
hostname(config-group-policy)# nac-sq-period seconds
hostname(config-group-policy)#
```

To inherit the value of the status query timer from the default group policy, access the alternative group policy from which to inherit it, then use the **no** form of this command:

```
hostname(config-group-policy)# no nac-sq-period [seconds]
hostname(config-group-policy)#
```

The following example changes the value of the status query timer to 1800 seconds:

```
hostname(config-group-policy)# nac-sq-period 1800
hostname(config-group-policy)
```

The following example inherits the value of the status query timer from the default group policy:

```
hostname(config-group-policy)# no nac-sq-period
hostname(config-group-policy)#
```

Step 2 (*Optional*) Configure the NAC revalidation period. The security appliance starts the revalidation timer after each successful posture validation. The expiration of this timer triggers the next unconditional posture validation. The security appliance maintains posture validation during revalidation. The default group policy becomes effective if the Access Control Server is unavailable during posture validation or revalidation. Enter the interval in seconds between each successful posture validation. The range is 300 through 86400. The default setting is 36000.

To specify the interval between each successful posture validation in a Network Admission Control session, use the **nac-reval-period** command in group-policy configuration mode:

```
hostname(config-group-policy)# nac-reval-period seconds
hostname(config-group-policy)#
```

To inherit the value of the Revalidation Timer from the default group policy, access the alternative group policy from which to inherit it, then use the **no** form of this command:

```
hostname(config-group-policy)# no nac-reval-period [seconds]
hostname(config-group-policy)#
```

The following example changes the revalidation timer to 86400 seconds:

```
hostname(config-group-policy)# nac-reval-period 86400
hostname(config-group-policy)
```

The following example inherits the value of the revalidation timer from the default group policy:

```
hostname(config-group-policy)# no nac-reval-period
hostname(config-group-policy)#
```

Step 3 (*Optional*) Configure the default ACL for NAC. The security appliance applies the security policy associated with the selected ACL if posture validation fails. Specify **none** or an extended ACL. The default setting is **none**. If the setting is **none** and posture validation fails, the security appliance applies the default group policy.

To specify the ACL to be used as the default ACL for Network Admission Control sessions that fail posture validation, use the **nac-default-acl** command in group-policy configuration mode:

```
hostname(config-group-policy)# nac-default-acl {acl-name | none}
hostname(config-group-policy)#
```

To inherit the ACL from the default group policy, access the alternative group policy from which to inherit it, then use the **no** form of this command:

```
hostname(config-group-policy)# no nac-default-acl [acl-name | none]
hostname(config-group-policy)#
```

The elements of this command are as follows:

- *acl-name*—Specifies the name of the posture validation server group, as configured on the security appliance using the **aaa-server host** command. The name must match the server-tag variable specified in that command.
- **none—**Disables inheritance of the ACL from the default group policy and does not apply an ACL to NAC sessions that fail posture validation.

Because NAC is disabled by default, VPN traffic traversing the security appliance is not subject to the NAC Default ACL until NAC is enabled.

The following example identifies acl-1 as the ACL to be applied when posture validation fails:

```
hostname(config-group-policy)# nac-default-acl acl-1
hostname(config-group-policy)
```

The following example inherits the ACL from the default group policy:

```
hostname(config-group-policy)# no nac-default-acl
hostname(config-group-policy)
```

The following example disables inheritance of the ACL from the default group policy and does not apply an ACL to NAC sessions that fail posture validation:

```
hostname(config-group-policy)# nac-default-acl none
hostname(config-group-policy)#
```

Step 4 Configure NAC exemptions for VPN. By default, the exemption list is empty.The default value of the filter attribute is **none**. Enter the **vpn-nac-exempt** once for each operating system (and ACL) to be matched to exempt remote hosts from posture validation.

To add an entry to the list of remote computer types that are exempt from posture validation, use the **vpn-nac-exempt** command in group-policy configuration mode.

```
hostname(config-group-policy)# vpn-nac-exempt os "os name" [filter {acl-name | none}]
[disable]
hostname(config-group-policy)#
```

To disable inheritance and specify that all hosts are subject to posture validation, use the **none** keyword immediately following **vpn-nac-exempt**.

```
hostname(config-group-policy)# vpn-nac-exempt none
hostname(config-group-policy)#
```

To remove an entry from the exemption list, use the **no** form of this command and name the operating system (and ACL) in the entry to be removed.

```
hostname(config-group-policy)# no vpn-nac-exempt [os "os name"] [filter {acl-name | none}]
[disable]
hostname(config-group-policy)#
```

To remove all entries from the exemption list associated with this group policy and inherit the list from the default group policy, use the **no** form of this command without specifying additional keywords.

```
hostname(config-group-policy)# no vpn-nac-exempt
hostname(config-group-policy)#
```

The syntax elements for these commands are as follows:

- *acl-name*—Name of the ACL present in the security appliance configuration.
- **disable—**Disables the entry in the exemption list without removing it from the list.
- **filter**—(*Optional*) filter to apply an ACL to filter the traffic if the computer matches the os name.

- **none—**When entered immediately after **vpn-nac-exempt**, this keyword disables inheritance and specifies that all hosts will be subject to posture validation.When entered immediately after **filter**, this keyword indicates that the entry does not specify an ACL.
- **OS—**Exempts an operating system from posture validation.
- *os name*—Operating system name. Quotation marks are required only if the name includes a space (for example, “Windows XP”).

The following example adds all hosts running Windows XP to the list of computers that are exempt from posture validation:

```
hostname(config-group-policy)# vpn-nac-exempt os "Windows XP"
hostname(config-group-policy)
```

The following example exempts all hosts running Windows 98 that match an ACE in the ACL named acl-1:

```
hostname(config-group-policy)# vpn-nac-exempt os "Windows 98" filter acl-1
hostname(config-group-policy)
```

The following example adds the same entry to the exemption list, but disables it:

```
hostname(config-group-policy)# vpn-nac-exempt os "Windows 98" filter acl-1 disable
hostname(config-group-policy)
```

The following example removes the same entry from the exemption list, regardless of whether it is disabled:

```
hostname(config-group-policy)# no vpn-nac-exempt os "Windows 98" filter acl-1
hostname(config-group-policy)
```

The following example disables inheritance and specifies that all hosts will be subject to posture validation:

```
hostname(config-group-policy)# no vpn-nac-exempt none
hostname(config-group-policy)
```

The following example removes all entries from the exemption list:

```
hostname(config-group-policy)# no vpn-nac-exempt
hostname(config-group-policy)
```

Step 5 Enable or disable Network Admission Control by entering the following command:

```
hostname(config-group-policy)# nac {enable | disable}
hostname(config-group-policy)#
```

To inherit the NAC setting from the default group policy, access the alternative group policy from which to inherit it, then use the **no** form of this command:

```
hostname(config-group-policy)# no nac [enable | disable]
hostname(config-group-policy)#
```

By default, NAC is disabled. Enabling NAC requires posture validation for remote access. If the remote computer passes the validation checks, the ACS server downloads the access policy for the security appliance to enforce. NAC is disabled by default.

An Access Control Server must be present on the network.

The following example enables NAC for the group policy:

```
hostname(config-group-policy)# nac enable
hostname(config-group-policy)#
```

Configuring Address Pools

Configure a list of address pools for allocating addresses to remote clients by entering the **address-pools** command in group-policy attributes configuration mode:

```
hostname(config-group-policy)# address-pools value address_pool1 [...address_pool6]
hostname(config-group-policy)#
```

The address-pools settings in this command override the local pool settings in the group. You can specify a list of up to six local address pools to use for local address allocation.

The order in which you specify the pools is significant. The security appliance allocates addresses from these pools in the order in which the pools appear in this command.

To remove the attribute from the group policy and enable inheritance from other sources of group policy, use the **no** form of this command:

```
hostname(config-group-policy)# no address-pools value address_pool1 [...address_pool6]
hostname(config-group-policy)#
```

The command **address-pools none** disables this attribute from being inherited from other sources of policy, such as the DefaultGrpPolicy:

```
hostname(config-group-policy)# address-pools none
hostname(config-group-policy)#
```

The command **no address pools none** removes the **address-pools none** command from the configuration, restoring the default value, which is to allow inheritance.

```
hostname(config-group-policy)# no address-pools none
hostname(config-group-policy)#
```

The syntax elements of this command are as follows:

- *address_pool*—Specifies the name of the address pool configured with the **ip local pool** command. You can specify up to 6 local address pools.
- **none—**Specifies that no address pools are configured and disables inheritance from other sources of group policy.
- **value—**Specifies a list of up to 6 address pools from which to assign addresses.

The following example entered in config-general configuration mode, configures pool 1 and pool20 as lists of address pools to use for allocating addresses to remote clients for GroupPolicy1:

```
hostname(config)# ip local pool pool 192.168.10.1-192.168.10.100 mask 255.255.0.0
hostname(config)# ip local pool pool20 192.168.20.1-192.168.20.200 mask 255.255.0.0
hostname(config)# group-policy GroupPolicy1 attributes
hostname(config-group-policy)# address-pools value pool1 pool20
hostname(config-group-policy)#
```

Configuring Firewall Policies

A *firewall* isolates and protects a computer from the Internet by inspecting each inbound and outbound individual packet of data to determine whether to allow or drop it. Firewalls provide extra security if remote users in a group have split tunneling configured. In this case, the firewall protects the user's PC,

and thereby the corporate network, from intrusions by way of the Internet or the user's local LAN. Remote users connecting to the security appliance with the VPN client can choose the appropriate firewall option.

Set personal firewall policies that the security appliance pushes to the VPN client during IKE tunnel negotiation by using the **client-firewall** command in group-policy configuration mode. To delete a firewall policy, enter the **no** form of this command.

To delete all firewall policies, enter the **no client-firewall** command without arguments. This command deletes all configured firewall policies, including a null policy if you created one by entering the **client-firewall** command with the **none** keyword.

When there are no firewall policies, users inherit any that exist in the default or other group policy. To prevent users from inheriting such firewall policies, enter the **client-firewall** command with the **none** keyword.

The Add or Edit Group Policy window, Client Firewall tab, lets you configure firewall settings for VPN clients for the group policy being added or modified.

Note Only VPN clients running Microsoft Windows can use these firewall features. They are currently not available to hardware clients or other (non-Windows) software clients.

In the first scenario, a remote user has a personal firewall installed on the PC. The VPN client enforces firewall policy defined on the local firewall, and it monitors that firewall to make sure it is running. If the firewall stops running, the VPN client drops the connection to the security appliance. (This firewall enforcement mechanism is called *Are You There (AYT)*, because the VPN client monitors the firewall by sending it periodic "are you there?" messages; if no reply comes, the VPN client knows the firewall is down and terminates its connection to the security appliance.) The network administrator might configure these PC firewalls originally, but with this approach, each user can customize his or her own configuration.

In the second scenario, you might prefer to enforce a centralized firewall policy for personal firewalls on VPN client PCs. A common example would be to block Internet traffic to remote PCs in a group using split tunneling. This approach protects the PCs, and therefore the central site, from intrusions from the Internet while tunnels are established. This firewall scenario is called *push policy* or *Central Protection Policy (CPP)*. On the security appliance, you create a set of traffic management rules to enforce on the VPN client, associate those rules with a filter, and designate that filter as the firewall policy. The security appliance pushes this policy down to the VPN client. The VPN client then in turn passes the policy to the local firewall, which enforces it.

Enter the following commands to set the appropriate client firewall parameters. You can configure only one instance of this command. Table 32-4, following this set of commands, explains the syntax elements of these commands:

Cisco Integrated Firewall

```
hostname(config-group-policy)# client-firewall {opt | req} cisco-integrated acl-in ACL
acl-out ACL
```

Cisco Security Agent

```
hostname(config-group-policy)# client-firewall {opt | req} cisco-security-agent
```

No Firewall

```
hostname(config-group-policy)# client-firewall none
```

Custom Firewall

```
hostname(config-group-policy)# client-firewall {opt | req} custom vendor-id num product-id
num policy {AYT | CPP acl-in ACL acl-out ACL} [description string]
```

Zone Labs Firewalls

```
hostname(config-group-policy)# client-firewall {opt | req} zonelabs-integrity
```

Note When the firewall type is **zonelabs-integrity**, do not include arguments. The Zone Labs Integrity Server determines the policies.

```
hostname(config-group-policy)# client-firewall {opt | req} zonelabs-zonealarm policy {AYT
| CPP acl-in ACL acl-out ACL}

hostname(config-group-policy)# client-firewall {opt | req} zonelabs-zonealarmorpro policy
{AYT | CPP acl-in ACL acl-out ACL}

client-firewall {opt | req} zonelabs-zonealarmpro policy {AYT | CPP acl-in ACL acl-out
ACL}
```

Sygate Personal Firewalls

```
hostname(config-group-policy)# client-firewall {opt | req} sygate-personal

hostname(config-group-policy)# client-firewall {opt | req} sygate-personal-pro

hostname(config-group-policy)# client-firewall {opt | req} sygate-security-agent
```

Network Ice, Black Ice Firewall:

```
hostname(config-group-policy)# client-firewall {opt | req} networkice-blackice
```

Table 32-4 client-firewall Command Keywords and Variables

Parameter	Description
acl-in *ACL*	Provides the policy the client uses for inbound traffic.
acl-out *ACL*	Provides the policy the client uses for outbound traffic.
AYT	Specifies that the client PC firewall application controls the firewall policy. The security appliance checks to make sure that the firewall is running. It asks, “Are You There?” If there is no response, the security appliance tears down the tunnel.
cisco-integrated	Specifies Cisco Integrated firewall type.
cisco-security-agent	Specifies Cisco Intrusion Prevention Security Agent firewall type.
CPP	Specifies Policy Pushed as source of the VPN client firewall policy.

Table 32-4 client-firewall Command Keywords and Variables

custom	Specifies Custom firewall type.
description *string*	Describes the firewall.
networkice-blackice	Specifies Network ICE Black ICE firewall type.
none	Indicates that there is no client firewall policy. Sets a firewall policy with a null value, thereby disallowing a firewall policy. Prevents inheriting a firewall policy from a default or specified group policy.
opt	Indicates an optional firewall type.
product-id	Identifies the firewall product.
req	Indicates a required firewall type.
sygate-personal	Specifies Sygate Personal firewall type.
sygate-personal-pro	Specifies Sygate Personal Pro firewall type.
sygate-security-agent	Specifies Sygate Security Agent firewall type.
vendor-id	Identifies the firewall vendor.
zonelabs-integrity	Specifies Zone Labs Integrity Server firewall type.
zonelabs-zonealarm	Specifies Zone Labs Zone Alarm firewall type.
zonelabs-zonealarmorpro policy	Specifies Zone Labs Zone Alarm or Pro firewall type.
zonelabs-zonealarmpro policy	Specifies Zone Labs Zone Alarm Pro firewall type.

The following example shows how to set a client firewall policy that requires Cisco Intrusion Prevention Security Agent for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# client-firewall req cisco-security-agent
hostname(config-group-policy)#
```

Configuring Client Access Rules

Configure rules that limit the remote access client types and versions that can connect via IPSec through the security appliance by using the **client-access-rule** command in group-policy configuration mode. Construct rules according to these guidelines:

- If you do not define any rules, the security appliance permits all connection types.
- When a client matches none of the rules, the security appliance denies the connection. If you define a deny rule, you must also define at least one permit rule; otherwise, the security appliance denies all connections.
- For both software and hardware clients, type and version must exactly match their appearance in the **show vpn-sessiondb remote** display.
- The * character is a wildcard, which you can enter multiple times in each rule. For example, **client-access rule 3 deny type * version 3.*** creates a priority 3 client access rule that denies all client types running release versions 3.x software.
- You can construct a maximum of 25 rules per group policy.
- There is a limit of 255 characters for an entire set of rules.

- You can enter n/a for clients that do not send client type and/or version.

To delete a rule, enter the **no** form of this command. This command is equivalent to the following command:

```
hostname(config-group-policy)# client-access-rule 1 deny type "Cisco VPN Client" version
4.0
```

To delete all rules, enter the **no client-access-rule command** without arguments. This deletes all configured rules, including a null rule if you created one by issuing the **client-access-rule** command with the **none** keyword.

By default, there are no access rules. When there are no client access rules, users inherit any rules that exist in the default group policy.

To prevent users from inheriting client access rules, enter the **client-access-rule** command with the **none** keyword. The result of this command is that all client types and versions can connect.

```
hostname(config-group-policy)# client-access rule priority {permit | deny} type type
version {version | none}

hostname(config-group-policy)# no client-access rule [priority {permit | deny} type type
version version]
```

Table 32-5 explains the meaning of the keywords and parameters in these commands.

Table 32-5 client-access rule Command Keywords and Variables

Parameter	Description
deny	Denies connections for devices of a particular type and/or version.
none	Allows no client access rules. Sets client-access-rule to a null value, thereby allowing no restriction. Prevents inheriting a value from a default or specified group policy.
permit	Permits connections for devices of a particular type and/or version.
priority	Determines the priority of the rule. The rule with the lowest integer has the highest priority. Therefore, the rule with the lowest integer that matches a client type and/or version is the rule that applies. If a lower priority rule contradicts, the security appliance ignores it.
type *type*	Identifies device types via free-form strings, for example VPN 3002. A string must match exactly its appearance in the **show vpn-sessiondb remote** display, except that you can enter the * character as a wildcard.
version *version*	Identifies the device version via free-form strings, for example 7.0. A string must match exactly its appearance in the **show vpn-sessiondb remote** display, except that you can enter the * character as a wildcard.

The following example shows how to create client access rules for the group policy named FirstGroup. These rules permit Cisco VPN clients running software version 4.x, while denying all Windows NT clients:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# client-access-rule 1 deny type WinNT version *
hostname(config-group-policy)# client-access-rule 2 permit "Cisco VPN Client" version 4.*
```

Note The "type" field is a free-form string that allows any value, but that value must match the fixed value that the client sends to the security appliance at connect time.

Configuring Group-Policy Attributes for Clientless SSL VPN Sessions

Clientless SSL VPN lets users establish a secure, remote-access VPN tunnel to the security appliance using a web browser. There is no need for either a software or hardware client. Clientless SSL VPN provides easy access to a broad range of web resources and web-enabled applications from almost any computer that can reach HTTPS Internet sites. Clientless SSL VPN uses SSL and its successor, TLS1, to provide a secure connection between remote users and specific, supported internal resources that you configure at a central site. The security appliance recognizes connections that need to be proxied, and the HTTP server interacts with the authentication subsystem to authenticate users. By default, clientless SSL VPN is disabled.

You can customize a configuration of clientless SSL VPN for specific internal group policies.

Note The webvpn mode that you enter from global configuration mode lets you configure global settings for clientless SSL VPN sessions. The webvpn mode described in this section, which you enter from group-policy configuration mode, lets you customize a configuration of group policies specifically for clientless SSL VPN sessions.

In group-policy webvpn configuration mode, you can specify whether to inherit or customize the following parameters, each of which is described in the subsequent sections:

- customizations
- html-content-filter
- homepage
- filter
- url-list
- port-forward
- port-forward-name
- sso server (single-signon server)
- auto-signon
- deny message
- SSL VPN Client (SVC)
- keep-alive ignore
- HTTP compression

In many instances, you define the webvpn attributes as part of configuring clientless SSL VPN, then you apply those definitions to specific groups when you configure the group-policy webvpn attributes. Enter group-policy webvpn configuration mode by using the **webvpn** command in group-policy configuration mode. Webvpn commands for group policies define access to files, URLs and TCP applications over clientless SSL VPN sessions. They also identify ACLs and types of traffic to filter. Clientless SSL VPN is disabled by default. See the description of *Chapter 39, "Configuring Clientless SSL VPN"* for more information about configuring the attributes for clientless SSL VPN sessions.

To remove all commands entered in group-policy webvpn configuration mode, enter the **no** form of this command. These webvpn commands apply to the username or group policy from which you configure them.

```
hostname(config-group-policy)# webvpn
hostname(config-group-policy)# no webvpn
```

The following example shows how to enter group-policy webvpn configuration mode for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)#
```

Applying Customization

Customizations determine the appearance of the windows that the user sees upon login. You configure the customization parameters as part of configuring clientless SSL VPN. To apply a previously defined web-page customization to change the look-and-feel of the web page that the user sees at login, enter the customization command in group-policy webvpn configuration mode:

```
hostname(config-group-webvpn)# customization customization_name
hostname(config-group-webvpn)#
```

For example, to use the customization named blueborder, enter the following command:

```
hostname(config-group-webvpn)# customization blueborder
hostname(config-group-webvpn)#
```

You configure the customization itself by entering the **customization** command in webvpn mode.

The following example shows a command sequence that first establishes a customization named 123 that defines a password prompt. The example then defines a group policy named testpolicy and uses the **customization** command to specify the use of the customization named 123 for clientless SSL VPN sessions:

```
hostname(config)# webvpn
hostname(config-webvpn)# customization 123
hostname(config-webvpn-custom)# password-prompt Enter password
hostname(config-webvpn)# exit
hostname(config)# group-policy testpolicy nopassword
hostname(config)# group-policy testpolicy attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# customization value 123
hostname(config-group-webvpn)#
```

Specifying a "Deny" Message

You can specify the message delivered to a remote user who logs into a clientless SSL VPN session successfully, but has no VPN privileges, by entering the **deny-message** command in group-policy webvpn configuration mode:

```
hostname(config-group-webvpn)# deny-message value "message"
hostname(config-group-webvpn)# no deny-message value "message"
hostname(config-group-webvpn)# deny-message none
```

The **no deny-message value** command removes the message string, so that the remote user does not receive a message.

The **no deny-message none** command removes the attribute from the connection profile policy configuration. The policy inherits the attribute value.

The message can be up to 491 alphanumeric characters long, including special characters, spaces, and punctuation, but not counting the enclosing quotation marks. The text appears on the remote user's browser upon login. When typing the string in the **deny-message value** command, continue typing even if the command wraps.

The default deny message is: "Login was successful, but because certain criteria have not been met or due to some specific group policy, you do not have permission to use any of the VPN features. Contact your IT administrator for more information."

The first command in the following example creates an internal group policy named group2. The subsequent commands modify the attributes, including the webvpn deny message associated with that policy.

```
hostname(config)# group-policy group2 internal
hostname(config)# group-policy group2 attributes
hostname(config-group)# webvpn
hostname(config-group-webvpn)# deny-message value "Your login credentials are OK. However,
you have not been granted rights to use the VPN features. Contact your administrator for
more information."
hostname(config-group-webvpn)
```

Configuring Group-Policy Filter Attributes for Clientless SSL VPN Sessions

Specify whether to filter Java, ActiveX, images, scripts, and cookies from clientless SSL VPN sessions for this group policy by using the **html-content-filter** command in webvpn mode. HTML filtering is disabled by default.

To remove a content filter, enter the **no** form of this command. To remove all content filters, including a null value created by issuing the **html-content-filter** command with the **none** keyword, enter the **no** form of this command without arguments. The **no** option allows inheritance of a value from another group policy. To prevent inheriting an html content filter, enter the **html-content-filter** command with the **none** keyword.

Using the command a second time overrides the previous setting.

```
hostname(config-group-webvpn)# html-content-filter {java | images | scripts | cookies |
none}

hostname(config-group-webvpn)# no html-content-filter [java | images | scripts | cookies |
none]
```

Table 32-6 describes the meaning of the keywords used in this command.

Table 32-6 filter Command Keywords

Keyword	Meaning
cookies	Removes cookies from images, providing limited ad filtering and privacy.
images	Removes references to images (removes <IMG> tags).
java	Removes references to Java and ActiveX (removes <EMBED>, <APPLET>, and <OBJECT> tags).
none	Indicates that there is no filtering. Sets a null value, thereby disallowing filtering. Prevents inheriting filtering values.
scripts	Removes references to scripting (removes <SCRIPT> tags).

The following example shows how to set filtering of JAVA and ActiveX, cookies, and images for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# html-content-filter java cookies images
hostname(config-group-webvpn)#
```

Specifying the User Home Page

Specify a URL for the web page that displays when a user in this group logs in by using the **homepage** command in group-policy webvpn configuration mode. There is no default home page.

To remove a configured home page, including a null value created by issuing the **homepage none** command, enter the **no** form of this command. The **no** option allows inheritance of a value from another group policy. To prevent inheriting a home page, enter the **homepage none** command.

The **none** keyword indicates that there is no home page for clientless SSL VPN sessions. It sets a null value, thereby disallowing a home page and prevents inheriting an home page.

The *url-string* variable following the keyword **value** provides a URL for the home page. The string must begin with either http:// or https://.

```
hostname(config-group-webvpn)# homepage {value url-string | none}
hostname(config-group-webvpn)# no homepage
hostname(config-group-webvpn)#
```

Configuring Auto-Signon

The **auto-signon** command is a single sign-on method for users of clientless SSL VPN sessions. It passes the login credentials (username and password) to internal servers for authentication using NTLM authentication, basic authentication, or both. Multiple auto-signon commands can be entered and are processed according to the input order (early commands take precedence).

You can use the auto-signon feature in three modes: webvpn configuration, webvpn group configuration, or webvpn username configuration mode. The typical precedence behavior applies where username supersedes group, and group supersedes global. The mode you choose depends upon the desired scope of authentication.

To disable auto-signon for a particular user to a particular server, use the **no** form of the command with the original specification of IP block or URI. To disable authentication to all servers, use the **no** form without arguments. The **no** option allows inheritance of a value from the group policy.

The following example, entered in group-policy webvpn configuration mode, configures auto-signon for the user named anyuser, using basic authentication, to servers with IP addresses ranging from 10.1.1.0 to 10.1.1.255:

The following example commands configure auto-signon for users of clientless SSL VPN sessions, using either basic or NTLM authentication, to servers defined by the URI mask https://*.example.com/*:

```
hostname(config)# group-policy ExamplePolicy attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# auto-signon allow uri https://*.example.com/* auth-type all
hostname(config-group-webvpn)#
```

The following example commands configure auto-signon for users of clientless SSL VPN sessions, using either basic or NTLM authentication, to the server with the IP address 10.1.1.0, using subnet mask 255.255.255.0:

```
hostname(config)# group-policy ExamplePolicy attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# auto-signon allow ip 10.1.1.0 255.255.255.0 auth-type all
```

```
hostname(config-group-webvpn)#
```

Specifying the Access List for Clientless SSL VPN Sessions

Specify the name of the access list to use for clientless SSL VPN sessions for this group policy or username by using the **filter** command in webvpn mode. Clientless SSL VPN access lists do not apply until you enter the **filter** command to specify them.

To remove the access list, including a null value created by issuing the **filter none** command, enter the **no** form of this command. The **no** option allows inheritance of a value from another group policy. To prevent inheriting filter values, enter the **filter value none** command.

Access lists for clientless SSL VPN sessions do not apply until you enter the **filter** command to specify them.

You configure ACLs to permit or deny various types of traffic for this group policy. You then enter the **filter** command to apply those ACLs for clientless SSL VPN traffic.

```
hostname(config-group-webvpn)# filter {value ACLname | none}
hostname(config-group-webvpn)# no filter
```

The **none** keyword indicates that there is no **webvpntype** access list. It sets a null value, thereby disallowing an access list and prevents inheriting an access list from another group policy.

The *ACLname* string following the keyword **value** provides the name of the previously configured access list.

Note Clientless SSL VPN sessions do not use ACLs defined in the **vpn-filter** command.

The following example shows how to set a filter that invokes an access list named *acl_in* for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# filter acl_in
hostname(config-group-webvpn)#
```

Applying a URL List

You can specify a list of URLs to appear on the clientless SSL VPN home page for a group policy. First, you must create one or more named lists by entering the **url-list** command in global configuration mode. To apply a list of servers and URLs for clientless SSL VPN sessions to a particular group policy, allowing access to the URLs in a list for a specific group policy, use the name of the list or lists you create there with the **url-list** command in group-policy webvpn configuration mode. There is no default URL list.

To remove a list, including a null value created by using the **url-list none** command**,** use the **no** form of this command. The **no** option allows inheritance of a value from another group policy. To prevent inheriting a URL list, use the **url-list none** command. Using the command a second time overrides the previous setting:

```
hostname(config-group-webvpn)# url-list {value name | none} [index]
hostname(config-group-webvpn)# no url-list
```

Table 32-7 shows the **url-list** command parameters and their meanings.

Table 32-7 url-list Command Keywords and Variables

Parameter	Meaning
index	Indicates the display priority on the home page.
none	Sets a null value for url lists. Prevents inheriting a list from a default or specified group policy.
value *name*	Specifies the name of a previously configured list of urls. To configure such a list, use the **url-list** command in global configuration mode.

The following example sets a URL list called FirstGroupURLs for the group policy named FirstGroup and specifies that this should be the first URL list displayed on the homepage:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# url-list value FirstGroupURLs 1
hostname(config-group-webvpn)#
```

Enabling ActiveX Relay for a Group Policy

ActiveX Relay lets a user who has established a Clientless SSL VPN session use the browser to launch Microsoft Office applications. The applications use the session to download and upload Microsoft Office documents. The ActiveX relay remains in force until the Clientless SSL VPN session closes.

To enable or disable ActiveX controls on Clientless SSL VPN sessions, enter the following command in group-policy webvpn configuration mode:

activex-relay {**enable** | **disable**}

To inherit the **activex-relay** command from the default group policy, enter the following command:

no activex-relay

The following commands enable ActiveX controls on clientless SSL VPN sessions associated with a given group policy:

```
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# activex-relay enable
hostname(config-group-webvpn)
```

Enabling Application Access on Clientless SSL VPN Sessions for a Group Policy

To enable application access for this group policy, enter the **port-forward** command in group-policy webvpn configuration mode. Port forwarding is disabled by default.

Before you can enter the **port-forward** command in group-policy webvpn configuration mode to enable application access, you must define a list of applications that you want users to be able to use in a clientless SSL VPN session. Enter the **port-forward** command in global configuration mode to define this list.

To remove the port forwarding attribute from the group-policy configuration, including a null value created by issuing the **port-forward none** command, enter the **no** form of this command. The **no** option allows inheritance of a list from another group policy. To prevent inheriting a port forwarding list, enter the **port-forward** command with the **none** keyword. The **none** keyword indicates that there is no filtering. It sets a null value, thereby disallowing a filtering, and prevents inheriting filtering values.

The syntax of the command is as follows:

```
hostname(config-group-webvpn)# port-forward {value listname | none}
hostname(config-group-webvpn)# no port-forward
```

The *listname* string following the keyword **value** identifies the list of applications users of clientless SSL VPN sessions can access. Enter the port-forward command in webvpn configuration mode to define the list.

Using the command a second time overrides the previous setting.

The following example shows how to set a port-forwarding list called *ports1* for the internal group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup internal attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# port-forward value ports1
hostname(config-group-webvpn)#
```

Configuring the Port-Forwarding Display Name

Configure the display name that identifies TCP port forwarding to end users for a particular user or group policy by using the **port-forward-name** command in group-policy webvpn configuration mode. To delete the display name, including a null value created by using the **port-forward-name none** command**,** enter the **no** form of the command. The **no** option restores the default name, Application Access. To prevent a display name, enter the **port-forward none** command. The syntax of the command is as follows:

```
hostname(config-group-webvpn)# port-forward-name {value name | none}
hostname(config-group-webvpn)# no port-forward-name
```

The following example shows how to set the name, Remote Access TCP Applications, for the internal group policy named *FirstGroup*:

```
hostname(config)# group-policy FirstGroup internal attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# port-forward-name value Remote Access TCP Applications
hostname(config-group-webvpn)#
```

Configuring the Maximum Object Size to Ignore for Updating the Session Timer

Network devices exchange short keepalive messages to ensure that the virtual circuit between them is still active. The length of these messages can vary. The **keep-alive-ignore** command lets you tell the security appliance to consider all messages that are less than or equal to the specified size as keepalive messages and not as traffic when updating the session timer. The range is 0 through 900 KB. The default is 4 KB.

To specify the upper limit of the HTTP/HTTPS traffic, per transaction, to ignore, use the **keep-alive-ignore** command in group-policy attributes webvpn configuration mode:

```
hostname(config-group-webvpn)# keep-alive-ignore size
hostname(config-group-webvpn)#

The no form of the command removes this specification from the configuration:
hostname(config-group-webvpn)# no keep-alive-ignore
hostname(config-group-webvpn)#
```

The following example sets the maximum size of objects to ignore as 5 KB:

```
hostname(config-group-webvpn)# keep-alive-ignore 5
hostname(config-group-webvpn)#
```

Specifying HTTP Compression

Enable compression of http data over a clientless SSL VPN session for a specific group or user by entering the **http-comp** command in the group policy webvpn mode.

```
hostname(config-group-webvpn)# http-comp {gzip | none}
hostname(config-group-webvpn)#
```

To remove the command from the configuration and cause the value to be inherited, use the **no** form of the command:

```
hostname(config-group-webvpn)# no http-comp {gzip | none}
hostname(config-group-webvpn)#
```

The syntax of this command is as follows:

- **gzip—**Specifies compression is enabled for the group or user. This is the default value.
- **none—**Specifies compression is disabled for the group or user.

For clientless SSL VPN sessions, the **compression** command configured from global configuration mode overrides the **http-comp** command configured in group policy and username webvpn modes.

In the following example, compression is disabled for the group-policy sales:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# http-comp none
hostname(config-group-webvpn)#
```

Specifying the SSO Server

Single sign-on support, available only for clientless SSL VPN sessions, lets users access different secure services on different servers without reentering a username and password more than once. The **sso-server value** command, when entered in group-policy-webvpn mode, lets you assign an SSO server to a group policy.

To assign an SSO server to a group policy, use the **sso-server value** command in group-policy-webvpn configuration mode. This command requires that your configuration include CA SiteMinder command.

```
hostname(config-group-webvpn)# sso-server value server_name
hostname(config-group-webvpn)#
```

To remove the assignment and use the default policy, use the **no** form of this command. To prevent inheriting the default policy, use the **sso-server none** command.

```
hostname(config-group-webvpn)# sso-server {value server_name | none}
hostname(config-group-webvpn)# [no] sso-server value server_name
```

The default policy assigned to the SSO server is DfltGrpPolicy.

The following example creates the group policy “my-sso-grp-pol” and assigns it to the SSO server named “example”:

```
hostname(config)# group-policy my-sso-grp-pol internal
hostname(config)# group-policy my-sso-grp-pol attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# sso-server value example
hostname(config-group-webvpn)#
```

Configuring SVC

The SSL VPN Client (SVC) is a VPN tunneling technology that gives remote users the benefits of an IPSec VPN client without the need for network administrators to install and configure IPSec VPN clients on remote computers. The SVC uses the SSL encryption that is already present on the remote computer as well as the clientless SSL VPN sessions login and authentication of the security appliance.

To establish an SVC session, the remote user enters the IP address of an interface of the security appliance configured to support clientless SSL VPN sessions. The browser connects to that interface and displays the clientless SSL VPN login screen. If the user satisfies the login and authentication, and the security appliance identifies the user as *requiring* the SVC, the security appliance downloads the SVC to the remote computer. If the security appliance identifies the user as having the *option* to use the SVC, the security appliance downloads the SVC to the remote computer while presenting a link on the user screen to skip the SVC installation.

After downloading, the SVC installs and configures itself, and then the SVC either remains or uninstalls itself (depending on the configuration) from the remote computer when the connection terminates.

The security appliance might have several unique SVC images residing in cache memory for different remote computer operating systems. When the user attempts to connect, the security appliance can consecutively download portions of these images to the remote computer until the image and operating system match, at which point it downloads the entire SVC. You can order the SVC images to minimize connection setup time, with the first image downloaded representing the most commonly-encountered remote computer operating system. For complete information about installing and using SVC, see Chapter 40, “Configuring AnyConnect VPN Client Connections”.

After enabling SVC, as described in Chapter 40, “Configuring AnyConnect VPN Client Connections”, you can enable or require SVC features for a specific group. This feature is disabled by default. If you enable or require SVC, you can then enable a succession of svc commands, described in this section. To enable SVC and its related svc commands, do the following steps in group-policy webvpn configuration mode:

Step 1 To enable the security appliance to download SVC files to remote computers, enter the **svc enable** command. By default, this command is disabled. The security appliance does not download SVC files. To remove the **svc enable** command from the configuration, use the **no** form of this command.

```
hostname(config-group-webvpn)# svc {none | enable | required}
hostname(config-group-webvpn)#
```

Note Entering the **no svc enable** command does not terminate active SVC sessions.

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# svc enable
hostname(config-group-webvpn)#
```

Step 2 To enable compression of HTTP data over an SVC connection, for a specific group, enter the svc compression command. By default, SVC compression is set to **deflate** (enabled). To disable compression for a specific group, use the **none** keyword. To remove the svc compression command and cause the value to be inherited, use the **no** form of the command:

```
hostname(config-group-webvpn)# svc compression {deflate | none}
hostname(config-group-webvpn)#
```

The following example disables SVC compression for the group policy named sales:

```
hostname(config)# group-policy sales attributes
```

```
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# svc compression none
hostname(config-group-webvpn)#
```

Step 3 To enable dead-peer-detection (DPD) on the security appliance and to set the frequency with which either the SVC or the security appliance performs DPD, use the **svc dpd-interval** command. To remove the **svc dpd-interval** command from the configuration, use the **no** form of the command. To disable SVC DPD for this group, use the **none** keyword:

```
hostname(config-group-webvpn)# svc dpd-interval {[gateway {seconds | none}] | [client
{seconds | none}]}
hostname(config-group-webvpn)#
```

DPD checking is disabled by default.

The gateway refers to the security appliance. You can specify the frequency with which the security appliance performs the DPD test as a range of from 30 to 3600 seconds (1 hour). Specifying **none** disables the DPD testing that the security appliance performs.

The client refers to the SVC. You can specify the frequency with which the client performs the DPD test as a range of from 30 to 3600 seconds (1 hour). Specifying **none** disables the DPD testing that the client performs.

In the following example, the user configures the DPD frequency performed by the security appliance (gateway) to 3000 seconds, and the DPD frequency performed by the client to 1000 seconds for the existing group policy named sales:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# svc dpd-interval gateway 3000
hostname(config-group-webvpn)# svc dpd-interval client 1000
hostname(config-group-webvpn)#
```

Step 4 You can adjust the frequency of keepalive messages (specified by *seconds)*, to ensure that an SVC connection through a proxy, firewall, or NAT device remains open, even if the device limits the time that the connection can be idle.

Adjusting the frequency also ensures that the SVC does not disconnect and reconnect when the remote user is not actively running a socket-based application, such as Microsoft Outlook or Microsoft Internet Explorer.

To configure the frequency (15 through 600 seconds) which an SVC on a remote computer sends keepalive messages to the security appliance, use the **svc keepalive** command. Use the **no** form of the command to remove the command from the configuration and cause the value to be inherited:

```
hostname(config-group-webvpn)# svc keepalive {none | seconds}
hostname(config-group-webvpn)# no svc keepalive {none | seconds}
hostname(config-group-webvpn)#
```

SVC keepalives are disabled by default. Using the keyword **none** disables SVC keepalive messages.

The following example configures the security appliance to enable the SVC to send keepalive messages, with a frequency of 300 seconds (5 minutes):

```
hostname(config-group-webvpn)# svc keepalive 300
hostname(config-group-webvpn)#
```

Step 5 To enable the permanent installation of an SVC onto a remote computer, use the **svc keep-installer** command with the **installed** keyword. To remove the command from the configuration, use the **no** form of this command:

```
hostname(config-group-webvpn)# svc keep-installer {installed | none}
hostname(config-group-webvpn)# no svc keep-installer {installed | none}
```

```
hostname(config-group-webvpn)#
```

The default is that permanent installation of the SVC is disabled. The SVC uninstalls from the remote computer at the end of the SVC session.

The following example configures the security appliance to keep the SVC installed on the remote computer for this group:

```
hostname(config-group-webvpn)# svc keep-installer installed
hostname(config-group-webvpn)#
```

Step 6 To enable the SVC to perform a rekey on an SVC session, use the **svc rekey** command. To disable rekey and remove the command from the configuration, use the **no** form of this command:

```
hostname(config-group-webvpn)# svc rekey {method {ssl | new-tunnel} | time minutes |
none}}
hostname(config-group-webvpn)# no svc rekey {method {ssl | new-tunnel} | time minutes |
none}}
hostname(config-group-webvpn)#
```

By default, SVC rekey is disabled.

Specifying the method as new-tunnel specifies that the SVC establishes a new tunnel during SVC rekey. Specifying the method as none disables SVC rekey. Specifying the method as ssl specifies that SSL renegotiation takes place during SVC rekey. instead of specifying the method, you can specify the time; that is, the number of minutes from the start of the session until the re-key takes place, from 1 through 10080 (1 week).

For the **no** form of the command, only the minimum is necessary, as the following example shows:

```
hostname(config-username-webvpn)# no svc rekey method
hostname(config-username-webvpn)#
```

If, however, you specify the method as new-tunnel:

```
hostname(config-username-webvpn)# no svc rekey method new-tunnel
hostname(config-username-webvpn)#
```

but the current method is ssl, then the command fails, because the values don't match.

In the following example, the user configures the SVC to renegotiate with SSL during rekey and configures the rekey to occur 30 minutes after the session begins:

```
hostname(config-group-webvpn)# svc rekey method ssl
hostname(config-group-webvpn)# svc rekey time 30
hostname(config-group-webvpn)#
```

Configuring User Attributes

This section describes user attributes and how to configure them. It includes the following sections:

- Viewing the Username Configuration, page 32-76
- Configuring Attributes for Specific Users, page 32-76

By default, users inherit all user attributes from the assigned group policy. The security appliance also lets you assign individual attributes at the user level, overriding values in the group policy that applies to that user. For example, you can specify a group policy giving all users access during business hours, but give a specific user 24-hour access.

Viewing the Username Configuration

To display the configuration for all usernames, including default values inherited from the group policy, enter the **all** keyword with the **show running-config username** command, as follows:

```
hostname# show running-config all username
hostname#
```

This displays the encrypted password and the privilege level. for all users, or, if you supply a username, for that specific user. If you omit the **all** keyword, only explicitly configured values appear in this list. The following example displays the output of this command for the user named testuser:

```
hostname# show running-config all username testuser
username testuser password 12RsxXQnphyr/I9Z encrypted privilege 15
```

Configuring Attributes for Specific Users

To configure specific users, you assign a password (or no password) and attributes to a user using the **username** command, which enters username mode. Any attributes that you do not specify are inherited from the group policy.

The internal user authentication database consists of the users entered with the **username** command. The **login** command uses this database for authentication. To add a user to the security appliance database, enter the **username** command in global configuration mode. To remove a user, use the **no** version of this command with the username you want to remove. To remove all usernames, use the **clear configure username** command without appending a username.

Setting a User Password and Privilege Level

Enter the **username** command to assign a password and a privilege level for a user. You can enter the **nopassword** keyword to specify that this user does not require a password. If you do specify a password, you can specify whether that password is stored in an encrypted form.

The optional **privilege** keyword lets you set a privilege level for this user. Privilege levels range from 0 (the lowest) through 15. System administrators generally have the highest privilege level. The default level is 2.

```
hostname(config)# username name {nopassword | password password [encrypted]} [privilege
priv_level]}

hostname(config)# no username [name]
```

Table 32-8 describes the meaning of the keywords and variables used in this command.

Table 32-8 username Command Keywords and Variables

Keyword/Variable	Meaning
encrypted	Indicates that the password is encrypted.
name	Provides the name of the user.
nopassword	Indicates that this user needs no password.

password *password*	Indicates that this user has a password, and provides the password.
privilege *priv_level*	Sets a privilege level for this user. The range is from 0 to 15, with lower numbers having less ability to use commands and administer the security appliance. The default privilege level is 2. The typical privilege level for a system administrator is 15.

By default, VPN users that you add with this command have no attributes or group policy association. You must explicitly configure all values.

The following example shows how to configure a user named anyuser with an encrypted password of pw_12345678 and a privilege level of 12:

```
hostname(config)# username anyuser password pw_12345678 encrypted privilege 12
hostname(config)#
```

Configuring User Attributes

After configuring the user's password (if any) and privilege level, you set the other attributes. These can be in any order. To remove any attribute-value pair, enter the **no** form of the command.

Enter username mode by entering the **username** command with the **attributes** keyword:

```
hostname(config)# username name attributes
hostname(config-username)#
```

The prompt changes to indicate the new mode. You can now configure the attributes.

Configuring VPN User Attributes

The VPN user attributes set values specific to VPN connections, as described in the following sections.

Configuring Inheritance

You can let users inherit from the group policy the values of attributes that you have not configured at the username level. To specify the name of the group policy from which this user inherits attributes, enter the **vpn-group-policy** command. By default, VPN users have no group-policy association:

```
hostname(config-username)# vpn-group-policy group-policy-name
hostname(config-username)# no vpn-group-policy group-policy-name
```

For an attribute that is available in username mode, you can override the value of an attribute in a group policy for a particular user by configuring it in username mode.

The following example shows how to configure a user named anyuser to use attributes from the group policy named FirstGroup:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-group-policy FirstGroup
hostname(config-username)#
```

Configuring Access Hours

Associate the hours that this user is allowed to access the system by specifying the name of a configured time-range policy:

To remove the attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a time-range value from another group policy. To prevent inheriting a value, enter the **vpn-access-hours none** command. The default is unrestricted access.

```
hostname(config-username)# vpn-access-hours value {time-range | none}
hostname(config-username)# vpn-access-hours value none
hostname(config)#
```

The following example shows how to associate the user named anyuser with a time-range policy called 824:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-access-hours 824
hostname(config-username)#
```

Configuring Maximum Simultaneous Logins

Specify the maximum number of simultaneous logins allowed for this user. The range is 0 through 2147483647. The default is 3 simultaneous logins. To remove the attribute from the running configuration, enter the **no** form of this command. Enter 0 to disable login and prevent user access.

```
hostname(config-username)# vpn-simultaneous-logins integer
hostname(config-username)# no vpn-simultaneous-logins
hostname(config-username)#
```

Note While the maximum limit for the number of simultaneous logins is very large, allowing several could compromise security and affect performance.

The following example shows how to allow a maximum of 4 simultaneous logins for the user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-simultaneous-logins 4
hostname(config-username)#
```

Configuring the Idle Timeout

Specify the idle timeout period in minutes, or enter **none** to disable the idle timeout. If there is no communication activity on the connection in this period, the security appliance terminates the connection.

The range is 1 through 35791394 minutes. The default is 30 minutes. To allow an unlimited timeout period, and thus prevent inheriting a timeout value, enter the **vpn-idle-timeout** command with the **none** keyword. To remove the attribute from the running configuration, enter the **no** form of this command.

```
hostname(config-username)# vpn-idle-timeout {minutes | none}
hostname(config-username)# no vpn-idle-timeout
hostname(config-username)#
```

The following example shows how to set a VPN idle timeout of 15 minutes for the user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-idle-timeout 30
hostname(config-username)#
```

Configuring the Maximum Connect Time

Specify the maximum user connection time in minutes, or enter **none** to allow unlimited connection time and prevent inheriting a value for this attribute. At the end of this period of time, the security appliance terminates the connection.

The range is 1 through 35791394 minutes. There is no default timeout. To allow an unlimited timeout period, and thus prevent inheriting a timeout value, enter the **vpn-session-timeout** command with the **none** keyword. To remove the attribute from the running configuration, enter the **no** form of this command.

```
hostname(config-username)# vpn-session-timeout {minutes | none}
hostname(config-username)# no vpn-session-timeout
hostname(config-username)#
```

The following example shows how to set a VPN session timeout of 180 minutes for the user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-session-timeout 180
hostname(config-username)#
```

Applying an ACL Filter

Specify the name of a previously-configured, user-specific ACL to use as a filter for VPN connections. To disallow an access list and prevent inheriting an access list from the group policy, enter the **vpn-filter** command with the none keyword. To remove the ACL, including a null value created by issuing the **vpn-filter none** command, enter the **no** form of this command. The **no** option allows inheritance of a value from the group policy. There are no default behaviors or values for this command.

You configure ACLs to permit or deny various types of traffic for this user. You then use the **vpn-filter** command to apply those ACLs.

```
hostname(config-username)# vpn-filter {value ACL_name | none}
hostname(config-username)# no vpn-filter
hostname(config-username)#
```

Note Clientless SSL VPN does not use ACLs defined in the **vpn-filter** command.

The following example shows how to set a filter that invokes an access list named acl_vpn for the user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-filter value acl_vpn
hostname(config-username)#
```

Specifying the IP Address and Netmask

Specify the IP address and netmask to assign to a particular user. To remove the IP address, enter the **no** form of this command.

```
hostname(config-username)# vpn-framed-ip-address {ip_address}
hostname(config-username)# no vpn-framed-ip-address
hostname(config-username)
```

The following example shows how to set an IP address of 10.92.166.7 for a user named anyuser:

```
hostname(config)# username anyuser attributes
```

```
hostname(config-username)# vpn-framed-ip-address 10.92.166.7
hostname(config-username)
```

Specify the network mask to use with the IP address specified in the previous step. If you used the **no vpn-framed-ip-address** command, do not specify a network mask. To remove the subnet mask, enter the **no** form of this command. There is no default behavior or value.

```
hostname(config-username)# vpn-framed-ip-netmask {netmask}
hostname(config-username)# no vpn-framed-ip-netmask
hostname(config-username)
```

The following example shows how to set a subnet mask of 255.255.255. 254 for a user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-framed-ip-netmask 255.255.255.254
hostname(config-username)
```

Specifying the Tunnel Protocol

Specify the VPN tunnel types (IPSec or clientless SSL VPN) that this user can use. The default is taken from the default group policy, the default for which is IPSec. To remove the attribute from the running configuration, enter the **no** form of this command.

```
hostname(config-username)# vpn-tunnel-protocol {webvpn | IPSec}
hostname(config-username)# no vpn-tunnel-protocol [webvpn | IPSec]
hostname(config-username)
```

The parameter values for this command are as follows:

- **IPSec—**Negotiates an IPSec tunnel between two peers (a remote access client or another secure gateway). Creates security associations that govern authentication, encryption, encapsulation, and key management.
- **webvpn—**Provides clientless SSL VPN access to remote users via an HTTPS-enabled web browser, and does not require a client

Enter this command to configure one or more tunneling modes. You must configure at least one tunneling mode for users to connect over a VPN tunnel.

The following example shows how to configure clientless SSL VPN and IPSec tunneling modes for the user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-tunnel-protocol webvpn
hostname(config-username)# vpn-tunnel-protocol IPSec
hostname(config-username)
```

Restricting Remote User Access

Configure the **group-lock** attribute with the **value** keyword to restrict remote users to access only through the specified, preexisting connection profile. Group-lock restricts users by checking whether the group configured in the VPN client is the same as the connection profile to which the user is assigned. If it is not, the security appliance prevents the user from connecting. If you do not configure group-lock, the security appliance authenticates users without regard to the assigned group.

To remove the **group-lock** attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a value from the group policy. To disable group-lock, and to prevent inheriting a group-lock value from a default or specified group policy, enter the **group-lock** command with the **none** keyword.

```
hostname(config-username)# group-lock {value tunnel-grp-name | none}
hostname(config-username)# no group-lock
hostname(config-username)
```

The following example shows how to set group lock for the user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# group-lock value tunnel-group-name
hostname(config-username)
```

Enabling Password Storage for Software Client Users

Specify whether to let users store their login passwords on the client system. Password storage is disabled by default. Enable password storage only on systems that you know to be in secure sites. To disable password storage, enter the **password-storage** command with the **disable** keyword. To remove the password-storage attribute from the running configuration, enter the **no** form of this command. This enables inheritance of a value for password-storage from the group policy.

```
hostname(config-username)# password-storage {enable | disable}
hostname(config-username)# no password-storage
hostname(config-username)
```

This command has no bearing on interactive hardware client authentication or individual user authentication for hardware clients.

The following example shows how to enable password storage for the user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# password-storage enable
hostname(config-username)
```

Configuring Clientless SSL VPN Access for Specific Users

The following sections describe how to customize a configuration for specific users of clientless SSL VPN sessions. Enter username webvpn configuration mode by using the **webvpn** command in username configuration mode. Clientless SSL VPN lets users establish a secure, remote-access VPN tunnel to the security appliance using a web browser. There is no need for either a software or hardware client. Clientless SSL VPN provides easy access to a broad range of web resources and web-enabled applications from almost any computer that can reach HTTPS Internet sites. Clientless SSL VPN uses SSL and its successor, TLS1, to provide a secure connection between remote users and specific, supported internal resources that you configure at a central site. The security appliance recognizes connections that need to be proxied, and the HTTP server interacts with the authentication subsystem to authenticate users.

The username webvpn configuration mode commands define access to files, URLs and TCP applications over clientless SSL VPN sessions. They also identify ACLs and types of traffic to filter. Clientless SSL VPN is disabled by default. These **webvpn** commands apply only to the username from which you configure them. Notice that the prompt changes, indicating that you are now in username webvpn configuration mode.

```
hostname(config-username)# webvpn
hostname(config-username-webvpn)#
```

To remove all commands entered in username webvpn configuration mode, use the **no** form of this command:

```
hostname(config-username)# no webvpn
hostname(config-username)#
```

You do not need to configure clientless SSL VPN to use e-mail proxies.

The security appliance does not support the Microsoft Outlook Exchange (MAPI) proxy. Neither port forwarding nor the smart tunnel feature that provides application access through a clientless SSL VPN session supports MAPI. For Microsoft Outlook Exchange communication using the MAPI protocol, remote users must use AnyConnect.



Note The webvpn mode that you enter from global configuration mode lets you configure global settings for clientless SSL VPN sessions. The username webvpn configuration mode described in this section, which you enter from username mode, lets you customize the configuration of specific users specifically for clientless SSL VPN sessions.

In username webvpn configuration mode, you can customize the following parameters, each of which is described in the subsequent steps:

- customizations
- deny message
- html-content-filter
- homepage
- filter
- url-list
- port-forward
- port-forward-name
- sso server (single-signon server)
- auto-signon
- SSL VPN Client (SVC)
- keep-alive ignore
- HTTP compression

The following example shows how to enter username webvpn configuration mode for the username anyuser attributes:

```
hostname(config)# username anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)#
```

Specifying the Content/Objects to Filter from the HTML

To filter Java, ActiveX, images, scripts, and cookies for clientless SSL VPN sessions for this user, enter the **html-content-filter** command in username webvpn configuration mode. To remove a content filter, enter the **no** form of this command. To remove all content filters, including a null value created by issuing the **html-content-filter none** command, enter the **no** form of this command without arguments. The **no** option allows inheritance of a value from the group policy. To prevent inheriting an HTML content filter, enter the **html-content-filter none** command. HTML filtering is disabled by default.

Using the command a second time overrides the previous setting.

```
hostname(config-username-webvpn)# html-content-filter {java | images | scripts | cookies |
none}
```

```
hostname(config-username-webvpn)# no html-content-filter [java | images | scripts |
cookies | none]
```

The keywords used in this command are as follows:

- **cookies—**Removes cookies from images, providing limited ad filtering and privacy.
- **images—**Removes references to images (removes <IMG> tags).
- **java—**Removes references to Java and ActiveX (removes <EMBED>, <APPLET>, and <OBJECT> tags.
- **none—**Indicates that there is no filtering. Sets a null value, thereby disallowing filtering. Prevents inheriting filtering values.
- **scripts**—Removes references to scripting (removes <SCRIPT> tags).

The following example shows how to set filtering of JAVA and ActiveX, cookies, and images for the user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# html-content-filter java cookies images
hostname(config-username-webvpn)#
```

Specifying the User Home Page

To specify a URL for the web page that displays when this user logs into clientless SSL VPN session, enter the **homepage** command in username webvpn configuration mode. To remove a configured home page, including a null value created by issuing the **homepage none** command, enter the **no** form of this command. The **no** option allows inheritance of a value from the group policy. To prevent inheriting a home page, enter the **homepage none** command.

The **none** keyword indicates that there is no clientless SSL VPN home page. It sets a null value, thereby disallowing a home page and prevents inheriting a home page.

The *url-string* variable following the keyword **value** provides a URL for the home page. The string must begin with either http:// or https://.

There is no default home page.

```
hostname(config-username-webvpn)# homepage {value url-string | none}
hostname(config-username-webvpn)# no homepage
hostname(config-username-webvpn)#
```

The following example shows how to specify www.example.com as the home page for the user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# homepage value www.example.com
hostname(config-username-webvpn)#
```

Applying Customization

Customizations determine the appearance of the windows that the user sees upon login. You configure the customization parameters as part of configuring clientless SSL VPN. To apply a previously defined web-page customization to change the look-and-feel of the web page that the user sees at login, enter the customization command in username webvpn configuration mode:

```
hostname(config-username-webvpn)# customization {none | value customization_name}
```

```
hostname(config-username-webvpn)#
```

For example, to use the customization named blueborder, enter the following command:

```
hostname(config-username-webvpn)# customization value blueborder
hostname(config-username-webvpn)#
```

You configure the customization itself by entering the **customization** command in webvpn mode.

The following example shows a command sequence that first establishes a customization named 123 that defines a password prompt. The example then defines a tunnel-group named test and uses the **customization** command to specify the use of the customization named 123:

```
hostname(config)# webvpn
hostname(config-webvpn)# customization 123
hostname(config-webvpn-custom)# password-prompt Enter password
hostname(config-webvpn)# exit
hostname(config)# username testuser nopassword
hostname(config)# username testuser attributes
hostname(config-username-webvpn)# webvpn
hostname(config-username-webvpn)# customization value 123
hostname(config-username-webvpn)#
```

Specifying a "Deny" Message

You can specify the message delivered to a remote user who logs into clientless SSL VPN session successfully, but has no VPN privileges by entering the **deny-message** command in username webvpn configuration mode:

```
hostname(config-username-webvpn)# deny-message value "message"
hostname(config-username-webvpn)# no deny-message value "message"
hostname(config-username-webvpn)# deny-message none
```

The **no deny-message value** command removes the message string, so that the remote user does not receive a message.

The **no deny-message none** command removes the attribute from the connection profile policy configuration. The policy inherits the attribute value.

The message can be up to 491 alphanumeric characters long, including special characters, spaces, and punctuation, but not counting the enclosing quotation marks. The text appears on the remote user's browser upon login. When typing the string in the **deny-message value** command, continue typing even if the command wraps.

The default deny message is: "Login was successful, but because certain criteria have not been met or due to some specific group policy, you do not have permission to use any of the VPN features. Contact your IT administrator for more information."

The first command in the following example enters username mode and configures the attributes for the user named anyuser. The subsequent commands enter username webvpn configuration mode and modify the deny message associated with that user.

```
hostname(config)# username anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# deny-message value "Your login credentials are OK.
However, you have not been granted rights to use the VPN features. Contact your
administrator for more information."
hostname(config-username-webvpn)
```

Specifying the Access List for Clientless SSL VPN Sessions

To specify the name of the access list to use for clientless SSL VPN sessions for this user, enter the **filter** command in username webvpn configuration mode. To remove the access list, including a null value created by issuing the **filter none** command, enter the **no** form of this command. The **no** option allows inheritance of a value from the group policy. To prevent inheriting filter values, enter the **filter value none** command.

Clientless SSL VPN access lists do not apply until you enter the **filter** command to specify them.

You configure ACLs to permit or deny various types of traffic for this user. You then enter the **filter** command to apply those ACLs for clientless SSL VPN traffic.

```
hostname(config-username-webvpn)# filter {value ACLname | none}
hostname(config-username-webvpn)# no filter
hostname(config-username-webvpn)#
```

The **none** keyword indicates that there is no **webvpntype** access list. It sets a null value, thereby disallowing an access list and prevents inheriting an access list from another group policy.

The *ACLname* string following the keyword **value** provides the name of the previously configured access list.

Note Clientless SSL VPN does not use ACLs defined in the **vpn-filter** command.

The following example shows how to set a filter that invokes an access list named *acl_in* for the user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# filter acl_in
hostname(config-username-webvpn)#
```

Applying a URL List

You can specify a list of URLs to appear on the home page for a user who has established a clientless SSL VPN session. First, you must create one or more named lists by entering the **url-list** command in global configuration mode. To apply a list of servers and URLs to a particular user of clientless SSL VPN, enter the **url-list** command in username webvpn configuration mode.

To remove a list, including a null value created by using the **url-list none** command**,** enter the **no** form of this command. The **no** option allows inheritance of a value from the group policy. To prevent inheriting a url list, enter the **url-list none** command.

```
hostname(config-username-webvpn)# url-list {listname displayname url | none}
hostname(config-username-webvpn)# no url-list
```

The keywords and variables used in this command are as follows:

- *displayname*—Specifies a name for the URL. This name appears on the portal page in the clientless SSL VPN session.
- *listname*—Identifies a name by which to group URLs.
- **none**—Indicates that there is no list of URLs. Sets a null value, thereby disallowing a URL list. Prevents inheriting URL list values.
- *url*—Specifies a URL that users of clientless SSL VPN can access.

There is no default URL list.

Using the command a second time overrides the previous setting.

The following example shows how to set a URL list called AnyuserURLs for the user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# url-list value AnyuserURLs
hostname(config-username-webvpn)#
```

Enabling ActiveX Relay for a User

ActiveX Relay lets a user who has established a Clientless SSL VPN session use the browser to launch Microsoft Office applications. The applications use the session to download and upload Microsoft Office documents. The ActiveX relay remains in force until the Clientless SSL VPN session closes.

To enable or disable ActiveX controls on Clientless SSL VPN sessions, enter the following command in username webvpn configuration mode:

activex-relay {**enable | disable**}

To inherit the **activex-relay** command from the group policy, enter the following command:

no activex-relay

The following commands enable ActiveX controls on Clientless SSL VPN sessions associated with a given username:

```
hostname(config-username-policy)# webvpn
hostname(config-username-webvpn)# activex-relay enable
hostname(config-username-webvpn)
```

Enabling Application Access for Clientless SSL VPN Sessions

To enable application access for this user, enter the **port-forward** command in username webvpn configuration mode. Port forwarding is disabled by default.

To remove the port forwarding attribute from the configuration, including a null value created by issuing the **port-forward none** command, enter the **no** form of this command. The **no** option allows inheritance of a list from the group policy. To disallow filtering and prevent inheriting a port forwarding list, enter the **port-forward** command with the **none** keyword.

```
hostname(config-username-webvpn)# port-forward {value listname | none}
hostname(config-username-webvpn)# no port-forward
hostname(config-username-webvpn)#
```

The *listname* string following the keyword **value** identifies the list of applications users of clientless SSL VPN can access. Enter the **port-forward** command in configuration mode to define the list.

Using the command a second time overrides the previous setting.

Before you can enter the **port-forward** command in username webvpn configuration mode to enable application access, you must define a list of applications that you want users to be able to use in a clientless SSL VPN session. Enter the **port-forward** command in global configuration mode to define this list.

The following example shows how to configure a portforwarding list called ports1:

```
hostname(config-group-policy)# webvpn
hostname(config-username-webvpn)# port-forward value ports1
hostname(config-username-webvpn)#
```

Configuring the Port-Forwarding Display Name

Configure the display name that identifies TCP port forwarding to end users for a particular user by using the **port-forward-name** command in username webvpn configuration mode. To delete the display name, including a null value created by using the **port-forward-name none** command**,** enter the **no** form of the command. The **no** option restores the default name, Application Access. To prevent a display name, enter the **port-forward none** command.

```
hostname(config-username-webvpn)# port-forward-name {value name | none}
hostname(config-username-webvpn)# no port-forward-name
```

The following example shows how to configure the port-forward name test:

```
hostname(config-group-policy)# webvpn
hostname(config-username-webvpn)# port-forward-name value test
hostname(config-username-webvpn)#
```

Configuring the Maximum Object Size to Ignore for Updating the Session Timer

Network devices exchange short keepalive messages to ensure that the virtual circuit between them is still active. The length of these messages can vary. The **keep-alive-ignore** command lets you tell the security appliance to consider all messages that are less than or equal to the specified size as keepalive messages and not as traffic when updating the session timer. The range is 0 through 900 KB. The default is 4 KB.

To specify the upper limit of the HTTP/HTTPS traffic, per transaction, to ignore, use the **keep-alive-ignore** command in group-policy attributes webvpn configuration mode:

```
hostname(config-group-webvpn)# keep-alive-ignore size
hostname(config-group-webvpn)#

The no form of the command removes this specification from the configuration:
hostname(config-group-webvpn)# no keep-alive-ignore
hostname(config-group-webvpn)#
```

The following example sets the maximum size of objects to ignore as 5 KB:

```
hostname(config-group-webvpn)# keep-alive-ignore 5
hostname(config-group-webvpn)#
```

Configuring Auto-Signon

To automatically submit the login credentials of a particular user of clientless SSL VPN to internal servers using NTLM, basic HTTP authentication or both, use the **auto-signon** command in username webvpn configuration mode.

The **auto-signon** command is a single sign-on method for users of clientless SSL VPN sessions. It passes the login credentials (username and password) to internal servers for authentication using NTLM authentication, basic authentication, or both. Multiple auto-signon commands can be entered and are processed according to the input order (early commands take precedence).

You can use the auto-signon feature in three modes: webvpn configuration, webvpn group configuration, or webvpn username configuration mode. The typical precedence behavior applies where username supersedes group, and group supersedes global. The mode you choose will depend upon the desired scope of authentication.

To disable auto-signon for a particular user to a particular server, use the **no** form of the command with the original specification of IP block or URI. To disable authentication to all servers, use the **no** form without arguments. The **no** option allows inheritance of a value from the group policy.

The following example commands configure auto-signon for a user of clientless SSL VPN named anyuser, using either basic or NTLM authentication, to servers defined by the URI mask https://*.example.com/*:

```
hostname(config)# username anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# auto-signon allow uri https://*.example.com/* auth-type
all
```

The following example commands configure auto-signon for a user of clientless SSL VPN named anyuser, using either basic or NTLM authentication, to the server with the IP address 10.1.1.0, using subnet mask 255.255.255.0:

```
hostname(config)# username anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# auto-signon allow ip 10.1.1.0 255.255.255.0 auth-type
all
hostname(config-username-webvpn)#
```

Specifying HTTP Compression

Enable compression of http data over a clientless SSL VPN session for a specific user by entering the **http-comp** command in the username webvpn configuration mode.

```
hostname(config-username-webvpn)# http-comp {gzip | none}
hostname(config-username-webvpn)#
```

To remove the command from the configuration and cause the value to be inherited, use the **no** form of the command:

```
hostname(config-username-webvpn)# no http-comp {gzip | none}
hostname(config-username-webvpn)#
```

The syntax of this command is as follows:

- **gzip—**Specifies compression is enabled for the group or user. This is the default value.
- **none—**Specifies compression is disabled for the group or user.

For clientless SSL VPN session, the **compression** command configured from global configuration mode overrides the **http-comp** command configured in group policy and username webvpn modes.

In the following example, compression is disabled for the username testuser:

```
hostname(config)# username testuser internal
hostname(config)# username testuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# http-comp none
hostname(config-username-webvpn)#
```

Specifying the SSO Server

Single sign-on support, available only for clientless SSL VPN sessions, lets users access different secure services on different servers without reentering a username and password more than once. The **sso-server value** command, when entered in username-webvpn mode, lets you assign an SSO server to a user.

To assign an SSO server to a user, use the **sso-server value** command in username-webvpn configuration mode. This command requires that your configuration include CA SiteMinder command.

```
hostname(config-username-webvpn)# sso-server value server_name
hostname(config-username-webvpn)#
```

To remove the assignment and use the default policy, use the **no** form of this command. To prevent inheriting the default policy, use the **sso-server none** command.

```
hostname(config-username-webvpn)# sso-server {value server_name | none}
hostname(config-username-webvpn)# [no] sso-server value server_name
```

The default policy assigned to the SSO server is DfltGrpPolicy.

The following example assigns the SSO server named example to the user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# sso-server value example
hostname(config-username-webvpn)#
```

Configuring SVC

The SSL VPN Client (SVC) is a VPN tunneling technology that gives remote users the benefits of an IPSec VPN client without the need for network administrators to install and configure IPSec VPN clients on remote computers. The SVC uses the SSL encryption that is already present on the remote computer as well as the login and authentication required to access the security appliance.

To establish an SVC session, the remote user enters the IP address of an interface of the security appliance configured to support clientless SSL VPN sessions. The browser connects to that interface and displays the login screen. If the user satisfies the login and authentication, and the security appliance identifies the user as *requiring* the SVC, the security appliance downloads the SVC to the remote computer. If the security appliance identifies the user as having the *option* to use the SVC, the security appliance downloads the SVC to the remote computer while presenting a link on the user screen to skip the SVC installation.

After downloading, the SVC installs and configures itself, and then the SVC either remains or uninstalls itself (depending on the configuration) from the remote computer when the connection terminates.

The security appliance might have several unique SVC images residing in cache memory for different remote computer operating systems. When the user attempts to connect, the security appliance can consecutively download portions of these images to the remote computer until the image and operating system match, at which point it downloads the entire SVC. You can order the SVC images to minimize connection setup time, with the first image downloaded representing the most commonly-encountered remote computer operating system. For complete information about installing and using SVC, see Chapter 40, “Configuring AnyConnect VPN Client Connections”.

After enabling SVC, as described in Chapter 40, “Configuring AnyConnect VPN Client Connections”, you can enable or require SVC features for a specific user. This feature is disabled by default. If you enable or require SVC, you can then enable a succession of svc commands, described in this section. To enable SVC and its related svc commands, do the following steps in username webvpn configuration mode:

Step 1 To enable the security appliance to download SVC files to remote computers, enter the **svc enable** command. By default, this command is disabled. The security appliance does not download SVC files. To remove the **svc enable** command from the configuration, use the **no** form of this command.

```
hostname(config-username-webvpn)# svc {none | enable | required}
hostname(config-username-webvpn)#
```



Note Entering the **no svc enable** command does not terminate active SVC sessions.

```
hostname(config)# username sales attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# svc enable
hostname(config-username-webvpn)#
```

Step 2 To enable compression of HTTP data over an SVC connection, for a specific user, enter the svc compression command. By default, SVC compression is set to **deflate** (enabled). To disable compression for a specific user, use the **none** keyword. To remove the svc compression command and cause the value to be inherited, use the **no** form of the command:

```
hostname(config-username-webvpn)# svc compression {deflate | none}
hostname(config-username-webvpn)#
```

The following example disables SVC compression for the user named sales:

```
hostname(config)# username sales attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# svc compression none
hostname(config-username-webvpn)#
```

Step 3 To enable dead-peer-detection (DPD) on the security appliance and to set the frequency with which either the SVC or the security appliance performs DPD, use the **svc dpd-interval** command. To remove the **svc dpd-interval** command from the configuration, use the **no** form of the command. To disable SVC DPD for this user, use the **none** keyword:

```
hostname(config-username-webvpn)# svc dpd-interval {[gateway {seconds | none}] | [client
{seconds | none}]}
hostname(config-username-webvpn)#
```

DPD checking is disabled by default.

The gateway refers to the security appliance. You can specify the frequency with which the security appliance performs the DPD test as a range of from 30 to 3600 seconds (1 hour). Specifying **none** disables the DPD testing that the security appliance performs.

The client refers to the SVC. You can specify the frequency with which the client performs the DPD test as a range of from 30 to 3600 seconds (1 hour). Specifying **none** disables the DPD testing that the client performs.

In the following example, the user configures the DPD frequency performed by the security appliance (gateway) to 3000 seconds, and the DPD frequency performed by the client to 1000 seconds for the existing user named sales:

```
hostname(config)# username sales attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# svc dpd-interval gateway 3000
hostname(config-username-webvpn)# svc dpd-interval client 1000
hostname(config-username-webvpn)#
```

Step 4 You can adjust the frequency of keepalive messages (specified by *seconds)*, to ensure that an SVC connection through a proxy, firewall, or NAT device remains open, even if the device limits the time that the connection can be idle.

Adjusting the frequency also ensures that the SVC does not disconnect and reconnect when the remote user is not actively running a socket-based application, such as Microsoft Outlook or Microsoft Internet Explorer.

To configure the frequency (15 through 600 seconds) which an SVC on a remote computer sends keepalive messages to the security appliance, use the **svc keepalive** command. Use the **no** form of the command to remove the command from the configuration and cause the value to be inherited:

```
hostname(config-username-webvpn)# svc keepalive {none | seconds}
hostname(config-username-webvpn)# no svc keepalive {none | seconds}
```

```
hostname(config-username-webvpn)#
```

SVC keepalives are disabled by default. Using the keyword **none** disables SVC keepalive messages.

In the following example, the user configures the security appliance to enable the SVC to send keepalive messages, with a frequency of 300 seconds (5 minutes):

```
hostname(config-username-webvpn)# svc keepalive 300
hostname(config-username-webvpn)#
```

Step 5 To enable the permanent installation of an SVC onto a remote computer, use the **svc keep-installer** command with the **installed** keyword. To remove the command from the configuration, use the **no** form of this command:

```
hostname(config-username-webvpn)# svc keep-installer {installed | none}
hostname(config-username-webvpn)# no svc keep-installer {installed | none}
hostname(config-username-webvpn)#
```

The default is that permanent installation of the SVC is disabled. The SVC uninstalls from the remote computer at the end of the SVC session.

The following example configures the security appliance to keep the SVC installed on the remote computer for this user:

```
hostname(config-username-webvpn)# svc keep-installer installed
hostname(config-username-webvpn)#
```

Step 6 To enable the SVC to perform a rekey on an SVC session, use the **svc rekey** command:

```
hostname(config-username-webvpn)# svc rekey {method {ssl | new-tunnel} | time minutes |
none}}
```

To disable rekey and remove the command from the configuration, use the **no** form of this command:

```
hostname(config-username-webvpn)# no svc rekey [method {ssl | new-tunnel} | time minutes |
none}]
hostname(config-username-webvpn)#
```

By default, SVC rekey is disabled.

Specifying the method as new-tunnel specifies that the SVC establishes a new tunnel during SVC rekey. Specifying the method as none disables SVC rekey. Specifying the method as ssl specifies that SSL renegotiation takes place during SVC rekey. instead of specifying the method, you can specify the time; that is, the number of minutes from the start of the session until the re-key takes place, from 1 through 10080 (1 week).

For the **no** form of the command, only the minimum is necessary. The following example is correct:

```
hostname(config-username-webvpn)# no svc rekey method
hostname(config-username-webvpn)#
```

If, however, you specify the method as new-tunnel:

```
hostname(config-username-webvpn)# no svc rekey method new-tunnel
hostname(config-username-webvpn)#
```

and the current method is ssl, then the command fails, because the values don't match.

In the following example, the user configures the SVC to renegotiate with SSL during rekey and configures the rekey to occur 30 minutes after the session begins:

```
hostname(config-username-webvpn)# svc rekey method ssl
hostname(config-username-webvpn)# svc rekey time 30
hostname(config-username-webvpn)#
```



CHAPTER 33

Configuring IP Addresses for VPNs

This chapter describes IP address assignment methods.

IP addresses make internetwork connections possible. They are like telephone numbers: both the sender and receiver must have an assigned number to connect. But with VPNs, there are actually two sets of addresses: the first set connects client and server on the public network. Once that connection is made, the second set connects client and server through the VPN tunnel.

In security appliance address management, we are dealing with the second set of IP addresses: those private IP addresses that connect a client with a resource on the private network, through the tunnel, and let the client function as if it were directly connected to the private network. Furthermore, we are dealing only with the private IP addresses that get assigned to clients. The IP addresses assigned to other resources on your private network are part of your network administration responsibilities, not part of VPN management. Therefore, when we discuss IP addresses here, we mean those IP addresses available in your private network addressing scheme that let the client function as a tunnel endpoint.

This chapter includes the following sections:

- Configuring an IP Address Assignment Method, page 33-1
- Configuring Local IP Address Pools, page 33-2
- Configuring AAA Addressing, page 33-2
- Configuring DHCP Addressing, page 33-3

Configuring an IP Address Assignment Method

The security appliance can use one or more of the following methods for assigning IP addresses to remote access clients. If you configure more than one address assignment method, the security appliance searches each of the options until it finds an IP address. By default, all methods are enabled. To view the current configuration, enter the **show running-config all vpn-addr-assign** command.

- **aaa**—Retrieves addresses from an external authentication server on a per-user basis. If you are using an authentication server that has IP addresses configured, we recommend using this method.
- **dhcp**—Obtains IP addresses from a DHCP server. If you want to use DHCP, you must configure a DHCP server. You must also define the range of IP addresses that the DHCP server can use.
- **local—Use an internal address pool.** Internally configured address pools are the easiest method of address pool assignment to configure. If you choose local, you must also use the **ip-local-pool** command to define the range of IP addresses to use.

To specify a method for assigning IP addresses to remote access clients, enter the **vpn-addr-assign** command in global configuration mode. The syntax is **vpn-addr-assign {aaa | dhcp | local}**.

Configuring Local IP Address Pools

To configure IP address pools to use for VPN remote access tunnels, enter the **ip local pool** command in global configuration mode. To delete address pools, enter the **no** form of this command.

The security appliance uses address pools based on the tunnel group for the connection. If you configure more than one address pool for a tunnel group, the security appliance uses them in the order in which they are configured.

If you assign addresses from a non-local subnet, we suggest that you add pools that fall on subnet boundaries to make adding routes for these networks easier.

A summary of the configuration of local address pools follows:

```
hostname(config)# vpn-addr-assign local
hostname(config)# ip local pool firstpool 10.20.30.40-10.20.30.50 mask 255.255.255.0
hostname(config)
```

Step 1 To configure IP address pools as the address assignment method, enter the **vpn-addr-assign** command with the **local** argument:

```
hostname(config)# vpn-addr-assign local
hostname(config)#
```

Step 2 To configure an address pool, enter the **ip local pool** command. **The syntax is ip local pool** *poolname first-address—last-address* **mask** *mask*.

The following example configures an IP address pool named firstpool. The starting address is 10.20.30.40 and the ending address is 10.20.30.50. The network mask is 255.255.255.0.

```
hostname(config)# ip local pool firstpool 10.20.30.40-10.20.30.50 mask 255.255.255.0
hostname(config)
```

Configuring AAA Addressing

To use a AAA server to assign addresses for VPN remote access clients, you must first configure a AAA server or server group. See the **aaa-server protocol** command in the *Cisco Security Appliance Command Reference* and "Identifying AAA Server Groups and Servers," in Chapter 14, "Configuring AAA Servers and the Local Database" of this guide.

In addition, the user must match a tunnel group configured for RADIUS authentication.

The following examples illustrate how to define a AAA server group called RAD2 for the tunnel group named firstgroup. It includes one more step than is necessary, in that previously you might have named the tunnel group and defined the tunnel group type. This step appears in the following example as a reminder that you have no access to subsequent tunnel-group commands until you set these values.

An overview of the configuration that these examples create follows:

```
hostname(config)# vpn-addr-assign aaa
hostname(config)# tunnel-group firstgroup type ipsec-ra
hostname(config)# tunnel-group firstgroup general-attributes
hostname(config-general)# authentication-server-group RAD2
```

To configure AAA for IP addressing, perform the following steps:

Step 1 To configure AAA as the address assignment method, enter the **vpn-addr-assign** command with the **aaa** argument:

```
hostname(config)# vpn-addr-assign aaa
hostname(config)#
```

Step 2 To establish the tunnel group called firstgroup as a remote access or LAN-to-LAN tunnel group, enter the **tunnel-group** command with the **type** keyword. The following example configures a remote access tunnel group.

```
hostname(config)# tunnel-group firstgroup type ipsec-ra
hostname(config)#
```

Step 3 To enter general-attributes configuration mode, which lets you define a AAA server group for the tunnel group called firstgroup, enter the **tunnel-group** command with the **general-attributes** argument.

```
hostname(config)# tunnel-group firstgroup general-attributes
hostname(config-general)#
```

Step 4 To specify the AAA server group to use for authentication, enter the **authentication-server-group** command.

```
hostname(config-general)# authentication-server-group RAD2
hostname(config-general)#
```

This command has more arguments that this example includes. For more information, see the *Cisco Security Appliance Command Reference*.

Configuring DHCP Addressing

To use DHCP to assign addresses for VPN clients, you must first configure a DHCP server and the range of IP addresses that the DHCP server can use. Then you define the DHCP server on a tunnel group basis. Optionally, you can also define a DHCP network scope in the group policy associated with the tunnel group or username. This is either an IP network number or IP Address that identifies to the DHCP server which pool of IP addresses to use.

The following examples define the DHCP server at IP address 172.33.44.19 for the tunnel group named firstgroup. They also define a DHCP network scope of 192.86.0.0 for the group policy called remotegroup. (The group policy called remotegroup is associated with the tunnel group called firstgroup). If you do not define a network scope, the DHCP server assigns IP addresses in the order of the address pools configured. It goes through the pools until it identifies an unassigned address.

The following configuration includes more steps than are necessary, in that previously you might have named and defined the tunnel group type as remote access, and named and identified the group policy as internal or external. These steps appear in the following examples as a reminder that you have no access to subsequent tunnel-group and group-policy commands until you set these values.

A summary of the configuration that these examples create follows:

```
hostname(config)# vpn-addr-assign dhcp
hostname(config)# tunnel-group firstgroup type ipsec-ra
hostname(config)# tunnel-group firstgroup general-attributes
hostname(config-general)# dhcp-server 172.33.44.19
hostname(config-general)# exit
hostname(config)# group-policy remotegroup internal
hostname(config)# group-policy remotegroup attributes
hostname(config-group-policy)# dhcp-network-scope 192.86.0.0
```

To define a DHCP server for IP addressing, perform the following steps.

Step 1 To configure DHCP as the address assignment method, enter the **vpn-addr-assign** command with the **dhcp** argument:

```
hostname(config)# vpn-addr-assign dhcp
hostname(config)#
```

Step 2 To establish the tunnel group called firstgroup as a remote access or LAN-to-LAN tunnel group, enter the **tunnel-group** command with the **type** keyword. The following example configures a remote access tunnel group.

```
hostname(config)# tunnel-group firstgroup type ipsec-ra
hostname(config)#
```

Step 3 To enter general-attributes configuration mode, which lets you configure a DHCP server, enter the **tunnel-group** command with the **general-attributes** argument.

```
hostname(config)# tunnel-group firstgroup general-attributes
hostname(config)#
```

Step 4 To define the DHCP server, enter the **dhcp-server** command. This command will allow you to configure the security appliance to send additional options to the specified DHCP servers when it is trying to get IP addresses for VPN clients. See the **dhcp-server** command in the *Cisco Security Appliance Command Reference* guide for more information. The following example configures a DHCP server at IP address 172.33.44.19.

```
hostname(config-general)# dhcp-server 172.33.44.19
hostname(config-general)#
```

Step 5 Exit tunnel-group mode.

```
hostname(config-general)# exit
hostname(config)#
```

Step 6 To define the group policy called remotegroup as an internally or externally configured group, enter the **group-policy** command with the **internal** or **external** argument. The following example configures an internal group.

```
hostname(config)# group-policy remotegroup internal
hostname(config)#
```

Step 7 (Optional) To enter group-policy attributes configuration mode, which lets you configure a subnetwork of IP addresses for the DHCP server to use, enter the **group-policy** command with the **attributes** keyword.

```
hostname(config)# group-policy remotegroup attributes
hostname(config-group-policy)#
```

Step 8 (Optional) To specify the range of IP addresses the DHCP server should use to assign addresses to users of the group policy called remotegroup, enter the dhcp-network-scope command. The following example configures at network scope of 192.86.0.0.

```
hostname(config-group-policy)# dhcp-network-scope 192.86.0.0
hostname(config-group-policy)#
```



CHAPTER 34

Configuring Remote Access IPSec VPNs

Remote access VPNs let single users connect to a central site through a secure connection over a TCP/IP network such as the Internet.

This chapter describes how to build a remote access VPN connection. It includes the following sections:

- Summary of the Configuration, page 34-1
- Configuring Interfaces, page 34-2
- Configuring ISAKMP Policy and Enabling ISAKMP on the Outside Interface, page 34-3
- Configuring an Address Pool, page 34-4
- Adding a User, page 34-4
- Creating a Transform Set, page 34-4
- Defining a Tunnel Group, page 34-5
- Creating a Dynamic Crypto Map, page 34-6
- Creating a Crypto Map Entry to Use the Dynamic Crypto Map, page 34-7

Summary of the Configuration

This chapter uses the following configuration to explain how to configure a remote access connection. Later sections provide step-by-step instructions.

```
hostname(config)# interface ethernet0
hostname(config-if)# ip address 10.10.4.200 255.255.0.0
hostname(config-if)# nameif outside
hostname(config-if)# no shutdown
hostname(config)# isakmp policy 1 authentication pre-share
hostname(config)# isakmp policy 1 encryption 3des
hostname(config)# isakmp policy 1 hash sha
hostname(config)# isakmp policy 1 group 2
hostname(config)# isakmp policy 1 lifetime 43200
hostname(config)# isakmp enable outside
hostname(config)# ip local pool testpool 192.168.0.10-192.168.0.15
hostname(config)# username testuser password 12345678
hostname(config)# crypto ipsec transform set FirstSet esp-3des esp-md5-hmac
hostname(config)# tunnel-group testgroup type ipsec-ra
hostname(config)# tunnel-group testgroup general-attributes
hostname(config-general)# address-pool testpool
hostname(config)# tunnel-group testgroup ipsec-attributes
hostname(config-ipsec)# pre-shared-key 44kkaol59636jnfx
hostname(config)# crypto dynamic-map dyn1 1 set transform-set FirstSet
```

```
hostname(config)# crypto dynamic-map dyn1 1 set reverse-route
hostname(config)# crypto map mymap 1 ipsec-isakmp dynamic dyn1
hostname(config)# crypto map mymap interface outside
hostname(config)# write memory
```

Configuring Interfaces

A security appliance has at least two interfaces, referred to here as outside and inside. Typically, the outside interface is connected to the public Internet, while the inside interface is connected to a private network and is protected from public access.

To begin, configure and enable two interfaces on the security appliance. Then assign a name, IP address and subnet mask. Optionally, configure its security level, speed and duplex operation on the security appliance.

To configure interfaces, perform the following steps, using the command syntax in the examples:

Step 1 To enter Interface configuration mode, in global configuration mode enter the **interface** command with the default name of the interface to configure. In the following example the interface is ethernet0.

```
hostname(config)# interface ethernet0
hostname(config-if)#
```

Step 2 To set the IP address and subnet mask for the interface, enter the **ip address** command. In the following example the IP address is 10.10.4.100 and the subnet mask is 255.255.0.0.

```
hostname(config-if)# ip address 10.10.4.200 255.255.0.0
hostname(config-if)#
```

Step 3 To name the interface, enter the **nameif** command, maximum of 48 characters. You cannot change this name after you set it. In the following example the name of the ethernet0 interface is outside.

```
hostname(config-if)# nameif outside
hostname(config-if)##
```

Step 4 To enable the interface, enter the **no** version of the **shutdown** command. By default, interfaces are disabled.

```
hostname(config-if)# no shutdown
hostname(config-if)#
```

Step 5 To save your changes, enter the **write memory** command.

```
hostname(config-if)# write memory
hostname(config-if)#
```

Step 6 To configure a second interface, use the same procedure.

Configuring ISAKMP Policy and Enabling ISAKMP on the Outside Interface

The Internet Security Association and Key Management Protocol, also called IKE, is the negotiation protocol that lets two hosts agree on how to build an IPSec Security Association. Each ISAKMP negotiation is divided into two sections called Phase1 and Phase2.

Phase 1 creates the first tunnel to protect later ISAKMP negotiation messages. Phase 2 creates the tunnel that protects data travelling across the secure connection.

To set the terms of the ISAKMP negotiations, you create an ISAKMP policy. It includes the following:

- An authentication method, to ensure the identity of the peers.
- An encryption method, to protect the data and ensure privacy.
- A Hashed Message Authentication Codes method to ensure the identity of the sender and to ensure that the message has not been modified in transit.
- A Diffie-Hellman group to set the size of the encryption key.
- A time limit for how long the security appliance uses an encryption key before replacing it.

See on page 29-3 in the "Configuring IPSec and ISAKMP" chapter of this guide for detailed information about the IKE policy keywords and their values.

To configure ISAKMP policies, in global configuration mode, enter the **isakmp policy** command with its various arguments. The syntax for ISAKMP policy commands is **isakmp policy** *priority* **attribute_name [attribute_value |** *integer***]**.

Perform the following steps and use the command syntax in the following examples as a guide.

Step 1 Set the authentication method. The following example configures preshared key. The priority is 1 in this and all following steps.

```
hostname(config)# isakmp policy 1 authentication pre-share
hostname(config)#
```

Step 2 Set the encryption method. The following example configures 3DES.

```
hostname(config)# isakmp policy 1 encryption 3des
hostname(config)#
```

Step 3 Set the HMAC method. The following example configures SHA-1.

```
hostname(config)# isakmp policy 1 hash sha
hostname(config)#
```

Step 4 Set the Diffie-Hellman group. The following example configures Group 2.

```
hostname(config)# isakmp policy 1 group 2
hostname(config)#
```

Step 5 Set the encryption key lifetime. The following example configures 43,200 seconds (12 hours).

```
hostname(config)# isakmp policy 1 lifetime 43200
hostname(config)#
```

Step 6 Enable ISAKMP on the interface named outside.

```
hostname(config)# isakmp enable outside
hostname(config)#
```

Step 7 To save your changes, enter the **write memory** command.

```
hostname(config)# write memory
hostname(config)#
```

Configuring an Address Pool

The security appliance requires a method for assigning IP addresses to users. A common method is using address pools. The alternatives are having a DHCP server assign address or having an AAA server assign them. The following example uses an address pool.

Step 1 To configure an address pool, enter the **ip local pool** command. The syntax is **ip local pool** *poolname first_address-last_address.* In the following example the pool name is testpool.

```
hostname(config)# ip local pool testpool 192.168.0.10-192.168.0.15
hostname(config)#
```

Step 2 Save your changes.

```
hostname(config)# write memory
hostname(config)#
```

Adding a User

To identify remote access users to the security appliance, configure usernames and passwords.

Step 1 To add users, enter the **username** command. The syntax is **username** *username* **password** *password*. In the following example the username is testuser and the password is 12345678.

```
hostname(config)# username testuser password 12345678
hostname(config)#
```

Step 2 Repeat Step 1 for each additional user.

Creating a Transform Set

A transform set combines an encryption method and an authentication method. During the IPSec security association negotiation with ISAKMP, the peers agree to use a particular transform set to protect a particular data flow. The transform set must be the same for both peers.

A transform set protects the data flows for the access list specified in the associated crypto map entry. You can create transform sets in the security appliance configuration, and then specify a maximum of 11 of them in a crypto map or dynamic crypto map entry. For more overview information, including a table that lists valid encryption and authentication methods, see Creating a Transform Set in Chapter 38, "Configuring LAN-to-LAN IPsec VPNs" of this guide.

Step 1 To configure a transform set, in global configuration mode enter the **crypto ipsec transform-set** command. The syntax is:

crypto ipsec transform-set *transform-set-name* ***encryption-method authentication-method***

The following example configures a transform set with the name FirstSet, esp-3des encryption, and esp-md5-hmac authentication:

```
hostname(config)# crypto ipsec transform set FirstSet esp-3des esp-md5-hmac
hostname(config)#
```

Step 2 Save the changes.

```
hostname(config)# write memory
hostname(config)#
```

Defining a Tunnel Group

A tunnel group is a set of records that contain tunnel connection policies. You configure a tunnel group to identify AAA servers, specify connection parameters, and define a default group policy. The security appliance stores tunnel groups internally.

There are two default tunnel groups in the security appliance system: DefaultRAGroup, which is the default IPSec remote-access tunnel group, and DefaultL2Lgroup, which is the default IPSec LAN-to-LAN tunnel group. You can change them but not delete them. The security appliance uses these groups to configure default tunnel parameters for remote access and LAN-to-LAN tunnel groups when there is no specific tunnel group identified during tunnel negotiation.

To establish a basic remote access connection, you must set three attributes for a tunnel group:

- Set the connection type to IPSec remote access.
- Configure the address assignment method, in the following example, address pool.
- Configure an authentication method, in the following example, preshared key.

Step 1 To set the connection type to IPSec remote access, enter the **tunnel-group** command. The command syntax is **tunnel-group** *name* **type** *type,* where *name* is the name you assign to the tunnel group, and *type* is the type of tunnel. The tunnel types as you enter them in the CLI include the following:

- ipsec-ra (IPSec remote access)
- ipsec-l2l (IPSec LAN to LAN)

In the following example the name of the tunnel group is testgroup.

```
hostname(config)# tunnel-group testgroup type ipsec-ra
hostname(config)#
```

Step 2 To configure an authentication method for the tunnel group, enter the general-attributes mode and then enter the **address-pool** command to create the address pool. In the following example the name of the group is testgroup and the name of the address pool is testpool.

```
hostname(config)# tunnel-group testgroup general-attributes
hostname(config-general)# address-pool testpool
```

Step 3 To configure the authentication method, enter the ipsec-attributes mode and then enter the **pre-shared-key** command to create the preshared key. You need to use the same preshared key on both the security appliance and the client.

Note The preshared key must be no larger than that used by the VPN client. If a Cisco VPN Client with a different preshared key size tries to connect to a security appliance, the client logs an error message indicating it failed to authenticate the peer.

The key is an alphanumeric string of 1-128 characters. In the following example the preshared key is 44kkaol59636jnfx.

```
hostname(config)# tunnel-group testgroup ipsec-attributes
hostname(config-ipsec)# pre-shared-key 44kkaol59636jnfx
```

Step 4 Save your changes.

```
hostname(config)# write memory
hostname(config)#
```

Creating a Dynamic Crypto Map

The security appliance uses dynamic crypto maps to define a policy template where all the parameters do not have to be configured. These dynamic crypto maps let the security appliance receive connections from peers that have unknown IP addresses. Remote access clients fall in this category.

Dynamic crypto map entries identify the transform set for the connection. You also enable reverse routing, which lets the security appliance learn routing information for connected clients, and advertise it via RIP or OSPF.

Step 1 To specify a transform set for a dynamic crypto map entry, enter the **crypto dynamic-map set transform-set** command.

The syntax is **crypto dynamic -map** *dynamic-map-name seq-num* **set transform-set** *transform-set-name*. In the following example the name of the dynamic map is dyn1, the sequence number is 1, and the transform set name is FirstSet.

```
hostname(config)# crypto dynamic-map dyn1 1 set transform-set FirstSet
hostname(config)#
```

Step 2 To enable RRI for any connection based on this crypto map entry, enter the **crypto dynamic-map set reverse route** command.

```
hostname(config)# crypto dynamic-map dyn1 1 set reverse-route
hostname(config)#
```

Step 3 Save your changes.

```
hostname(config)# write memory
hostname(config)#
```

Creating a Crypto Map Entry to Use the Dynamic Crypto Map

Next create a crypto map entry that lets the security appliance use the dynamic crypto map to set the parameters of IPSec security associations.

In the following examples for this command, the name of the crypto map is mymap, the sequence number is 1, and the name of the dynamic crypto map is dyn1, which you created in the previous section, "Creating a Dynamic Crypto Map." Enter these commands in global configuration mode.

Step 1 To create a crypto map entry that uses a dynamic crypto map, enter the **crypto map** command. The syntax is **crypto map** *map-name seq-num* **ipsec-isakmp dynamic** *dynamic-map-name.*

```
hostname(config)# crypto map mymap 1 ipsec-isakmp dynamic dyn1
hostname(config)#
```

Step 2 To apply the crypto map to the outside interface, enter the **crypto map interface** command.

The syntax is **crypto map** *map-name* **interface** *interface-name*

```
hostname(config)# crypto map mymap interface outside
hostname(config)#
```



CHAPTER 35

Configuring Network Admission Control

This chapter includes the following sections:

- Overview, page 35-1
- Uses, Requirements, and Limitations, page 35-2
- Viewing the NAC Policies on the Security Appliance, page 35-2
- Adding, Accessing, or Removing a NAC Policy, page 35-4
- Configuring a NAC Policy, page 35-4
- Assigning a NAC Policy to a Group Policy, page 35-8
- Changing Global NAC Framework Settings, page 35-8

Overview

Network Admission Control protects the enterprise network from intrusion and infection from worms, viruses, and rogue applications by performing endpoint compliancy and vulnerability checks as a condition for production access to the network. We refer to these checks as *posture validation.* You can configure posture validation to ensure that the anti-virus files, personal firewall rules, or intrusion protection software on a host with an IPSec or WebVPN session are up-to-date before providing access to vulnerable hosts on the intranet. Posture validation can include the verification that the applications running on the remote hosts are updated with the latest patches. NAC occurs only after user authentication and the setup of the tunnel. NAC is especially useful for protecting the enterprise network from hosts that are not subject to automatic network policy enforcement, such as home PCs.

The establishment of a tunnel between the endpoint and the security appliance triggers posture validation.

You can configure the security appliance to pass the IP address of the client to an optional audit server if the client does not respond to a posture validation request. The audit server, such as a Trend server, uses the host IP address to challenge the host directly to assess its health. For example, it may challenge the host to determine whether its virus checking software is active and up-to-date. After the audit server completes its interaction with the remote host, it passes a token to the posture validation server, indicating the health of the remote host.

Following successful posture validation or the reception of a token indicating the remote host is healthy, the posture validation server sends a network access policy to the security appliance for application to the traffic on the tunnel.

In a *NAC Framework* configuration involving the security appliance, only a Cisco Trust Agent running on the client can fulfill the role of posture agent, and only a Cisco Access Control Server (ACS) can fulfill the role of posture validation server. The ACS uses dynamic ACLs to determine the access policy for each client.

As a RADIUS server, the ACS can authenticate the login credentials required to establish a tunnel, in addition to fulfilling its role as posture validation server.

Note Only a NAC Framework policy configured on the security appliance supports the use of an audit server.

In its role as posture validation server, the ACS uses access control lists. If posture validation succeeds and the ACS specifies a redirect URL as part of the access policy it sends to the security appliance, the security appliance redirects all HTTP and HTTPS requests from the remote host to the redirect URL. Once the posture validation server uploads an access policy to the security appliance, all of the associated traffic must pass both the Security Appliance and the ACS (or vice versa) to reach its destination.

The establishment of a tunnel between an IPSec or WebVPN client and the security appliance triggers posture validation if a NAC Framework policy is assigned to the group policy. The NAC Framework policy can, however, identify operating systems that are exempt from posture validation and specify an optional ACL to filter such traffic.

Uses, Requirements, and Limitations

When configured to support NAC, the security appliance functions as a client of a Cisco Secure Access Control Server, requiring that you install a minimum of one Access Control Server on the network to provide NAC authentication services.

Following the configuration of one or more Access Control Servers on the network, you must use the **aaa-server** command to name the Access Control Server group. Then follow the instructions in the "Configuring a NAC Policy" procedure on page 35-4.

ASA support for NAC Framework is limited to remote access IPSec and WebVPN client sessions. The NAC Framework configuration supports only single mode.

NAC on the ASA does not support Layer 3 (non-VPN) traffic and IPv6 traffic.

Viewing the NAC Policies on the Security Appliance

Before configuring the NAC policies to be assigned to group policies, we recommend that you view any that may already be set up on the security appliance. To do so, enter the following command in privileged EXEC mode:

show running-config nac-policy

The default configuration does not contain NAC policies, however, entering this command is a useful way to determine whether anyone has added any. If so, you may decide that the policies already configured are suitable and disregard the section on configuring a NAC policy.

The following example shows the configuration of a NAC policy named nacframework1:

```
hostname# show running-config nac-policy
nac-policy nacframework1 nac-framework
```

```
 default-acl acl-1
 reval-period 36000
 sq-period 300
 exempt-list os "Windows XP" filter acl-2
hostname#
```

The first line of each NAC policy indicates its name and type (nac-framework). Table 35-1 explains the nac-framework attributes displayed in response to the **show running-config nac-policy** command.

Table 35-1 show running-config nac-policy Command Fields

Field	Description
default-acl	NAC default ACL applied before posture validation. Following posture validation, the security appliance replaces the default ACL with the one obtained from the Access Control Server for the remote host. The security appliance retains the default ACL if posture validation fails.
reval-period	Number of seconds between each successful posture validation in a NAC Framework session.
sq-period	Number of seconds between each successful posture validation in a NAC Framework session and the next query for changes in the host posture
exempt-list	Operating system names that are exempt from posture validation. Also shows an optional ACL to filter the traffic if the remote computer's operating system matches the name.
authentication-server-group	name of the of authentication server group to be used for NAC posture validation.

To display the assignment of NAC policies to group policies, enter the following command in privileged EXEC mode:

show nac-policy

In addition to listing the NAC policy-to-group policy assignments, the CLI shows which NAC policies are unassigned and the usage count for each NAC policy, as follows:

```
asa2(config)# show nac-policy
nac-policy framework1 nac-framework
  applied session count = 0
  applied group-policy count = 2
  group-policy list:    GroupPolicy2    GroupPolicy1
nac-policy framework2 nac-framework is not in use.
asa2(config)#
```

The CLI shows the text "is not in use" next to the policy type if the policy is not assigned to any group policies. Otherwise, the CLI displays the policy name and type on the first line and the usage data for the group policies in subsequent lines. Table 35-2 explains the fields in the **show nac-policy** command.

Table 35-2 show nac-policy Command Fields

Field	Description
applied session count	Cumulative number of VPN sessions to which this security appliance applied the NAC policy.
applied group-policy count	Cumulative number of group polices to which this security appliance applied the NAC policy.
group-policy list	List of group policies to which this NAC policy is assigned. In this case, the usage of a group policy does not determine whether it appears in this list; if the NAC policy is assigned to a group policy in the running configuration, then the group policy appears in this list.

Refer to the following sections to create a NAC policy or modify one that is already present.

Adding, Accessing, or Removing a NAC Policy

Enter the following command in global configuration mode to add or modify a NAC policy:

[**no**] **nac-policy** *nac-policy-name* **nac-framework**

Use the **no** version of the command to remove a NAC policy from the configuration. Alternatively, you can enter the **clear configure nac-policy** command to remove all NAC policies from the configuration except for those that are assigned to group policies. When entering the command to remove or prepare to modify a NAC policy, you must specify both the name and type of the policy.

nac-policy-name is the name of a new NAC policy or one that is already present. The name is a string of up to 64 characters. The **show running-config nac-policy** command displays the name and configuration of each NAC policy already present on the security appliance.

nac-framework specifies that a NAC Framework configuration will provide a network access policy for remote hosts. A Cisco Access Control Server must be present on the network to provide NAC Framework services for the security appliance. When you specify this type, the prompt indicates you are in `nac-policy-nac-framework` configuration mode. This mode lets you configure the NAC Framework policy.

You can create more than one NAC Framework policy, but you can assign no more than one to a group policy.

For example, the following command creates and accesses a NAC Framework policy named nac-framework1:

```
hostname(config)# nac-policy nac-framework1 nac-framework
hostname(config-nac-policy-nac-framework)
```

Configuring a NAC Policy

After you use the **nac-policy** command to name a NAC Framework policy, use the following sections to assign values to its attributes before you assign it to a group policy.

Specifying the Access Control Server Group

You must configure at least one Cisco Access Control Server to support NAC. Use the **aaa-server host** command to name the Access Control Server group even if the group contains only one server.

You can enter the following command to display the AAA server configuration:

show running-config aaa-server

For example:

```
hostname(config)# show running-config aaa-server
aaa-server acs-group1 protocol radius
aaa-server acs-group1 (outside) host 192.168.22.44
 key secret
 radius-common-pw secret
hostname(config)#
```

Enter the following command in nac-policy-nac-framework configuration mode to specify the group to be used for NAC posture validation:

[**no**] **authentication-server-group** *server-group*

Use the **no** form of the command if you want to remove the command from the NAC policy.

server-group must match the server-tag variable specified in the **aaa-server host** command. It is optional if you are using the **no** version of the command.

For example, enter the following command to specify acs-group1 as the authentication server group to be used for NAC posture validation:

```
hostname(config-nac-policy-nac-framework)# authentication-server-group acs-group1
hostname(config-nac-policy-nac-framework)
```

Setting the Query-for-Posture-Changes Timer

After each successful posture validation, the security appliance starts a status query timer. The expiration of this timer triggers a query to the remote host for changes in posture since the last posture validation. A response indicating no change resets the status query timer. A response indicating a change in posture triggers an unconditional posture revalidation. The security appliance maintains the current access policy during revalidation.

By default, the interval between each successful posture validation and the status query, and each subsequent status query, is 300 seconds (5 minutes). Enter the following command in nac-policy-nac-framework configuration mode to change the status query interval:

[**no**] **sq-period** *seconds*

Use the **no** form of the command if you want to turn off the status query timer. If you turn off this timer and enter **show running-config nac-policy**, the CLI displays a `0` next to the `sq-period` attribute, which means the timer is turned off.

seconds must be in the range 30 to 1800 seconds (5 to 30 minutes). It is optional if you are using the **no** version of the command.

The following example changes the status query timer to 1800 seconds:

```
hostname(config-group-policy)# sq-period 1800
hostname(config-group-policy)
```

Setting the Revalidation Timer

After each successful posture validation, the security appliance starts a revalidation timer. The expiration of this timer triggers the next unconditional posture validation. The security appliance maintains the current access policy during revalidation.

By default, the interval between each successful posture validation is 36000 seconds (10 hours). To change it, enter the following command in nac-policy-nac-framework configuration mode:

[**no**] **reval-period** *seconds*

Use the **no** form of the command if you want to turn off the status query timer. If you turn off this timer and enter **show running-config nac-policy**, the CLI displays a `0` next to the `sq-period` attribute, which means the timer is turned off.

seconds must be in the range is 300 to 86400 seconds (5 minutes to 24 hours). It is optional if you are using the **no** version of the command.

For example, enter the following command to change the revalidation timer to 86400 seconds:

```
hostname(config-nac-policy-nac-framework)# reval-period 86400
hostname(config-nac-policy-nac-framework)
```

Configuring the Default ACL for NAC

Each group policy points to a default ACL to be applied to hosts that match the policy and are eligible for NAC. The security appliance applies the NAC default ACL before posture validation. Following posture validation, the security appliance replaces the default ACL with the one obtained from the Access Control Server for the remote host. The security appliance retains the default ACL if posture validation fails.

The security appliance also applies the NAC default ACL if clientless authentication is enabled (which is the default setting).

Enter the following command in nac-policy-nac-framework configuration mode to specify the ACL to be used as the default ACL for NAC sessions:

[**no**] **default-acl** *acl-name*

Use the **no** form of the command if you want to remove the command from the NAC Framework policy. In that case, specifying the *acl-name* is optional.

acl-name is the name of the access control list to be applied to the session.

The following example identifies acl-2 as the ACL to be applied before posture validation succeeds:

```
hostname(config-nac-policy-nac-framework)# default-acl acl-2
hostname(config-nac-policy-nac-framework)
```

Configuring Exemptions from NAC

The security appliance configuration stores a list of exemptions from NAC posture validation. You can specify the operating systems that are exempt. If you specify an ACL, the client running the operating system specified is exempt from posture validation and the client traffic is subject to the ACL.

To add an entry to the list of remote computer types that are exempt from NAC posture validation, enter the following command in nac-policy-nac-framework configuration mode:

[no] exempt-list os "*os-name*" [**disable** | **filter** *acl-name* [**disable**]]

The **no exempt-list** command removes all exemptions from the NAC Framework policy. Specifying an entry when issuing the **no** form of the command removes the entry from the exemption list.

Note When the command specifies an operating system, it does not overwrite the previously added entry to the exception list; enter the command once for each operating system and ACL you want to exempt.

os exempts an operating system from posture validation.

os-name is the operating system name. Use quotation marks if the name includes a space (for example, “Windows XP”).

filter applies an ACL to filter the traffic if the computer’s operating system matches the *os name*. The **filter/***acl-name* pair is optional.

disable performs one of two functions, as follows:

- If you enter it after the "os-name," the security appliance ignores the exemption, and applies NAC posture validation to the remote hosts that are running that operating system.
- If you enter it after the *acl-name*, security appliance exempts the operating system, but does not apply the ACL to the associated traffic.

acl-name is the name of the ACL present in the security appliance configuration. When specified, it must follow the **filter** keyword.

For example, enter the following command to add all hosts running Windows XP to the list of computers that are exempt from posture validation:

```
hostname(config-group-policy)# exempt-list os "Windows XP"
hostname(config-group-policy)
```

The following example exempts all hosts running Windows XP and applies the ACL acl-2 to traffic from those hosts:

```
hostname(config-nac-policy-nac-framework)# exempt-list os "Windows XP" filter acl-2
hostname(config-nac-policy-nac-framework)
```

The following example removes the same entry from the exemption list:

```
hostname(config-nac-policy-nac-framework)# no exempt-list os "Windows XP" filter acl-2
hostname(config-nac-policy-nac-framework)
```

The following example removes all entries from the exemption list:

```
hostname(config-nac-policy-nac-framework)# no exempt-list
hostname(config-nac-policy-nac-framework)
```

Assigning a NAC Policy to a Group Policy

Upon completion of each tunnel setup, the security appliance applies the NAC policy, if it is assigned to the group policy, to the session.

To assign a NAC policy to a group policy, use the **nac-settings** command in group-policy configuration mode, as follows:

[**no**] **nac-settings** { **value** *nac-policy-name* | **none** }

no nac-settings removes the *nac-policy-name* from the group policy. The group policy inherits the nac-settings value from the default group policy.

nac-settings none removes the *nac-policy-name* from the group policy and disables the use of a NAC policy for this group policy. The group policy does not inherit the nac-settings value from the default group policy.

nac-settings value assigns the NAC policy you name to the group policy. To display the name and configuration of each NAC policy, enter the **show running-config nac-policy** command.

By default, the **nac-settings** command is not present in the configuration of each group policy. The security appliance automatically enables NAC for a group policy when you assign a NAC policy to it.

The following example command assigns the NAC policy named framework1 to the group policy:

```
hostname(config-group-policy)# nac-settings value framework1
hostname(config-group-policy)
```

Changing Global NAC Framework Settings

The security appliance provides default settings for a NAC Framework configuration. Use the instructions in this section to adjust these settings for adherence to the policies in force in your network.

Changing Clientless Authentication Settings

NAC Framework support for clientless authentication is configurable. It applies to hosts that do not have a Cisco Trust Agent to fulfill the role of posture agent. The security appliance applies the default access policy, sends the EAP over UDP request for posture validation, and the request times out. If the security appliance is not configured to request a policy for clientless hosts from the Access Control Server, it retains the default access policy already in use for the clientless host. If the security appliance is configured to request a policy for clientless hosts from the Access Control Server, it does so and the Access Control Server downloads the access policy to be enforced by the security appliance.

Enabling and Disabling Clientless Authentication

Enter the following command in global configuration mode to enable clientless authentication for a NAC Framework configuration:

[**no**] **eou allow** {**audit** | **clientless** | **none**}

audit uses an audit server to perform clientless authentication.

clientless uses a Cisco Access Control Server to perform clientless authentication.

no removes the command from the configuration.

none disables clientless authentication.

The default configuration contains the **eou allow clientless** configuration.

Note The **eou** commands apply *only* to NAC Framework sessions.

Clientless authentication is enabled by default.

The following example shows how to configure the security appliance to use an audit server to perform clientless authentication:

```
hostname(config)# eou allow audit
hostname(config)#
```

The following example shows how to disable the use of an audit server:

```
hostname(config)# no eou allow audit
hostname(config)#
```

Changing the Login Credentials Used for Clientless Authentication

When clientless authentication is enabled, and the security appliance fails to receive a response to a validation request from the remote host, it sends a clientless authentication request on behalf of the remote host to the Access Control Server. The request includes the login credentials that match those configured for clientless authentication on the Access Control Server. The default username and password for clientless authentication on the security appliance matches the default username and password on the Access Control Server; the default username and password are both “clientless”. If you change these values on the Access Control Server, you must also do so on the security appliance.

Enter the following command in global configuration mode to change the username used for clientless authentication:

eou clientless username *username*

username must match the username configured on the Access Control Server to support clientless hosts. Enter 1 to 64 ASCII characters, excluding leading and trailing spaces, pound signs (#), question marks (?), quotation marks ("), asterisks (*), and angle brackets (< and >).

Enter the following command in global configuration mode to change the password used for clientless authentication:

eou clientless password *password*

password must match the password configured on the Access Control Server to support clientless hosts. Enter 4 – 32 ASCII characters.

You can specify only the username, only the password, or both. For example, enter the following commands to change the username and password for clientless authentication to sherlock and 221B-baker, respectively:

```
hostname(config)# eou clientless username sherlock
hostname(config)# eou clientless password 221B-baker
hostname(config)#
```

To change the username to its default value, enter the following command:

no eou clientless username

For example:

```
hostname(config)# no eou clientless username
hostname(config)#
```

To change the password to its default value, enter the following command:

no eou clientless password

For example:

```
hostname(config)# no eou clientless password
hostname(config)#
```

Changing NAC Framework Session Attributes

The ASA provides default settings for the attributes that specify communications between the security appliance and the remote host. These attributes specify the port no. to communicate with posture agents on remote hosts and the expiration counters that impose limits on the communications with the posture agents. These attributes, the default settings, and the commands you can enter to change them are as follows:

- Port no. on the client endpoint to be used for EAP over UDP communication with posture agents.

  The default port no. is 21862. Enter the following command in global communication mode to change it:

  eou port *port_number*

  port_number must match the port number configured on the CTA. Enter a value in the range 1024 to 65535.

  For example, enter the following command to change the port number for EAP over UDP communication to 62445:

  ```
  hostname(config)# eou port 62445
  hostname(config)#
  ```

  To change the port number to its default value, use the **no** form of this command, as follows:

  no eou port

  For example:

  ```
  hostname(config)# no eou port
  hostname(config)#
  ```

- Retransmission retry timer

  When the security appliance sends an EAP over UDP message to the remote host, it waits for a response. If it fails to receive a response within *n* seconds, it resends the EAP over UDP message. By default, the retransmission timer is 3 seconds. To change this value, enter the following command in global configuration mode:

  eou timeout retransmit *seconds*

seconds is a value in the range 1 to 60.

The following example changes the retransmission timer to 6 seconds:

```
hostname(config)# eou timeout retransmit 6
hostname(config)#
```

To change the retransmission retry timer to its default value, use the **no** form of this command, as follows:

no eou timeout **retransmit**

For example:

```
hostname(config)# no eou timeout retransmit
hostname(config)#
```

- Retransmission retries

  When the security appliance sends an EAP over UDP message to the remote host, it waits for a response. If it fails to receive a response, it resends the EAP over UDP message. By default, it retries up to 3 times. To change this value, enter the following command in global configuration mode:

  eou max-retry *retries*

  retries is a value in the range 1 to 3.

  The following example limits the number of EAP over UDP retransmissions to 1:

  ```
  hostname(config)# eou max-retry 1
  hostname(config)#
  ```

  To change the maximum number of retransmission retries to its default value, use the **no** form of this command, as follows:

  no eou max-retry

  For example:

  ```
  hostname(config)# no eou max-retry
  hostname(config)#
  ```

- Session reinitialization timer

  When the retransmission retry counter matches the max-retry value, the security appliance terminates the EAP over UDP session with the remote host and starts the hold timer. When the hold timer equals *n* seconds, the security appliance establishes a new EAP over UDP session with the remote host. By default, the maximum number of seconds to wait before establishing a new session is 180 seconds. To change this value, enter the following command in global configuration mode:

  eou timeout hold-period *seconds*

  seconds is a value in the range 60 to 86400.

  For example, enter the following command to change the wait period before initiating a new EAP over UDP association to 120 seconds:

  ```
  hostname(config)# eou timeout hold-period 120
  hostname(config)#
  ```

  To change the session reinitialization to its default value, use the **no** form of this command, as follows:

no eou timeout hold-period

For example:

```
hostname(config)# no eou timeout hold-period
hostname(config)#
```



CHAPTER 36

Configuring Easy VPN Services on the ASA 5505

This chapter describes how to configure the ASA 5505 as an Easy VPN hardware client. This chapter assumes you have configured the switch ports and VLAN interfaces of the ASA 5505 (see Chapter 5, "Configuring Switch Ports and VLAN Interfaces for the Cisco ASA 5505 Adaptive Security Appliance").

Note The Easy VPN hardware client configuration specifies the IP address of its primary and secondary (backup) Easy VPN servers. Any ASA, including another ASA 5505 configured as a headend, a VPN 3000 Series Concentrator, an IOS-based router, or a firewall can act as an Easy VPN server. An ASA 5505 cannot, however function as both a client and a server simultaneously. To configure an ASA 5505 as a server, see "Specifying the Client/Server Role of the Cisco ASA 5505" section on page 36-1. Then configure the ASA 5505 as you would any other ASA, beginning with the "Getting Started" section on page 2-1 of this guide.

This chapter includes the following sections:

- Specifying the Client/Server Role of the Cisco ASA 5505, page 36-1
- Specifying the Primary and Secondary Servers, page 36-2
- Specifying the Mode, page 36-3
- Configuring Automatic Xauth Authentication, page 36-4
- Configuring IPSec Over TCP, page 36-4
- Comparing Tunneling Options, page 36-5
- Specifying the Tunnel Group or Trustpoint, page 36-6
- Configuring Split Tunneling, page 36-8
- Configuring Device Pass-Through, page 36-8
- Configuring Remote Management, page 36-9
- Guidelines for Configuring the Easy VPN Server, page 36-9

Specifying the Client/Server Role of the Cisco ASA 5505

The Cisco ASA 5505 can function as a Cisco Easy VPN hardware client (also called "Easy VPN Remote") or as a server (also called a "headend"), but not both at the same time. It does not have a default role. Use one of the following commands in global configuration mode to specify its role:

- vpnclient enable to specify the role of the ASA 5505 as an Easy VPN Remote
- **no vpnclient enable** to specify the role of the ASA 5505 as server

The following example shows how to specify the ASA 5505 as an Easy VPN hardware client:

```
hostname(config)# vpnclient enable
hostname(config)#
```

The CLI responds with an error message indicating that you must remove certain data elements if you switch from server to hardware client, depending on whether the elements are present in the configuration. Table 36-1 lists the data elements that are permitted in both client and server configurations, and not permitted in client configurations.

Table 36-1 Configuration Privileges and Restrictions on the ASA 5505

Permitted in Both Client and Server Configurations	Not Permitted in Client Configurations
crypto ca trustpoints	exempting MAC addreses from authentication and authorization (aaa mac-exempt command)
digital certificates	tunnel groups
group policies	ISAKMP policies
crypto dynamic-maps	IPsec over TCP
crypto ipsec transform-sets	crypto maps
crypto ipsec security-association lifetime	virtual HTTP/HTTPs
crypto ipsec fragmentation before-encryption	AAA authentication rules
crypto ipsec df-bit copy-df	failover
	global NAT/PAT pools
	SSL VPN, both clientless and AnyConnect

An ASA 5505 configured as an Easy VPN hardware client retains the commands listed in the first column within its configuration, however, some have no function in the client role.

The following example shows how to specify the ASA 5505 as an Easy VPN server:

```
hostname(config)# no vpnclient enable
hostname(config)#
```

After entering the no version of this command, configure the ASA 5505 as you would any other ASA, beginning with "Getting Started" section on page 2-1 of this guide.

Specifying the Primary and Secondary Servers

Before establishing a connection with an Easy VPN hardware client, you must specify the IP address of an Easy VPN server to which it will connect. Any ASA, including another ASA 5505 configured as a headend, a VPN 3000 Series Concentrator, an IOS-based router, or a firewall can act as an Easy VPN server. Use the **vpnclient server** command in global configuration mode, as follows:

[**no**] **vpnclient server** *ip_primary* [*ip_secondary_1…ip_secondary_10*]

no removes the command from the running configuration.

ip_primary_address is the IP address or DNS name of the primary Easy VPN server.

ip_secondary_address_n (Optional) is a list of the IP addresses or DNS names of up to ten backup Easy VPN servers. Use a space to separate the items in the list.

For example, enter the following command to configure a VPN client to use Easy VPN Server 10.10.10.15 as the primary server, and 10.10.10.30 and 192.168.10.45 as alternate servers:

```
hostname(config)# vpnclient server 10.10.10.15 10.10.10.30 192.168.10.10
hostname(config)#
```

Specifying the Mode

The Easy VPN Client supports one of two modes of operation: Client Mode or Network Extension Mode (NEM). The mode of operation determines whether the inside hosts relative to the Easy VPN Client are accessible from the Enterprise network over the tunnel. Specifying a mode of operation is mandatory before making a connection because Easy VPN Client does not have a default mode.

Client mode, also called Port Address Translation (PAT) mode, isolates the IP addresses of all devices on the Easy VPN Client private network from those on the enterprise network. The Easy VPN Client performs PAT for all VPN traffic for its inside hosts. IP address management is neither required for the Easy VPN Client inside interface or the inside hosts.

NEM makes the inside interface and all inside hosts routeable across the enterprise network over the tunnel. Hosts on the inside network obtain their IP addresses from an accessible subnet (statically or via DHCP) pre-configured with static IP addresses. PAT does not apply to VPN traffic in NEM. This mode does not require a VPN configuration for each client. The Cisco ASA 5505 configured for NEM mode supports automatic tunnel initiation. The configuration must store the group name, user name, and password. Automatic tunnel initiation is disabled if secure unit authentication is enabled.

Note If the Easy VPN hardware client is using NEM and has connections to secondary servers, use the **crypto map set reverse-route** command on each headend device to configure dynamic announcements of the remote network using Reverse Route Injection (RRI).

To specify the mode for Easy VPN Clients, enter the following command in configuration mode:

[**no**] **vpnclient mode** {**client-mode** | **network-extension-mode**}

no removes the command from the running configuration.

NEM with Multiple Interfaces

If you have an ASA 5505 security appliance (version 7.2 (3) and higher) configured as an Easy VPN Client in Network Extension Mode with multiple interfaces configured, the security appliance builds a tunnel for locally encrypted traffic only from the interface with the highest security level.

For example, consider the following configuration:

```
vlan1 security level 100 nameif inside
vlan2 security level 0 nameif outside
vlan12 security level 75 nameif work
```

In this scenario, the security appliance builds the tunnel only for vlan1, the interface with the highest security level. If you want to encrypt traffic from vlan12, you must change the security level of interface vlan1 to a lower value than that of vlan 12.

Configuring Automatic Xauth Authentication

The ASA 5505 configured as an Easy VPN hardware client automatically authenticates when it connects to the Easy VPN server if all of the following conditions are true:

- Secure unit authentication is disabled on the server.
- The server requests IKE Extended Authenticate (Xauth) credentials.

  Xauth provides the capability of authenticating a user within IKE using TACACS+ or RADIUS. Xauth authenticates a user (in this case, the Easy VPN hardware client) using RADIUS or any of the other supported user authentication protocols.
- The client configuration contains an Xauth username and password.

Enter the following command in global configuration mode to configure the Xauth username and password:

vpnclient username *xauth_username* **password** *xauth password*

You can use up to 64 characters for each.

For example, enter the following command to configure the Easy VPN hardware client to use the XAUTH username testuser and password ppurkm1:

```
hostname(config)# vpnclient username testuser password ppurkm1
hostname(config)#
```

To remove the username and password from the running configuration, enter the following command:

no vpnclient username

For example:

```
hostname(config)# no vpnclient username
hostname(config)#
```

Configuring IPSec Over TCP

By default, the Easy VPN hardware client and server encapsulate IPSec in User Datagram Protocol (UDP) packets. Some environments, such as those with certain firewall rules, or NAT and PAT devices, prohibit UDP. To use standard Encapsulating Security Protocol (ESP, Protocol 50) or Internet Key Exchange (IKE, UDP 500) in such environments, you must configure the client and the server to encapsulate IPSec within TCP packets to enable secure tunneling. If your environment allows UDP, however, configuring IPSec over TCP adds unnecessary overhead.

To configure the Easy VPN hardware client to use TCP-encapsulated IPSec, enter the following command in global configuration mode:

vpnclient ipsec-over-tcp [**port** *tcp_port*]

The Easy VPN hardware client uses port 10000 if the command does not specify a port number.

If you configure an ASA 5505 to use TCP-encapsulated IPSec, enter the following command to let it send large packets over the outside interface:

```
hostname(config)# crypto ipsec df-bit clear-df outside
hostname(config)#
```

This command clears the Don't Fragment (DF) bit from the encapsulated header. A DF bit is a bit within the IP header that determines whether the packet can be fragmented. This command lets the Easy VPN hardware client send packets that are larger than the MTU size.

The following example shows how to configure the Easy VPN hardware client to use TCP-encapsulated IPSec, using the default port 10000, and to let it send large packets over the outside interface:

```
hostname(config)# vpnclient ipsec-over-tcp
hostname(config)# crypto ipsec df-bit clear-df outside
hostname(config)#
```

The next example shows how to configure the Easy VPN hardware client to use TCP-encapsulated IPSec, using the port 10501, and to let it send large packets over the outside interface:

```
hostname(config)# vpnclient ipsec-over-tcp port 10501
hostname(config)# crypto ipsec df-bit clear-df outside
hostname(config)#
```

To remove the attribute from the running configuration, use the **no** form of this command, as follows:

no vpnclient ipsec-over-tcp

For example:

```
hostname(config)# no vpnclient ipsec-over-tcp
hostname(config)#
```

Comparing Tunneling Options

The tunnel types the Cisco ASA 5505 configured as an Easy VPN hardware client sets up depends on a combination of the following factors:

- Use of the **split-tunnel-network-list** and the **split-tunnel-policy** commands on the headend to permit, restrict, or prohibit split tunneling. (See the Creating a Network List for Split-Tunneling, page 32-48 and "Setting the Split-Tunneling Policy" section on page 32-47, respectively.)

  Split tunneling determines the networks for which the remote-access client encrypts and sends data through the secured VPN tunnel, and determines which traffic it sends to the Internet in the clear.

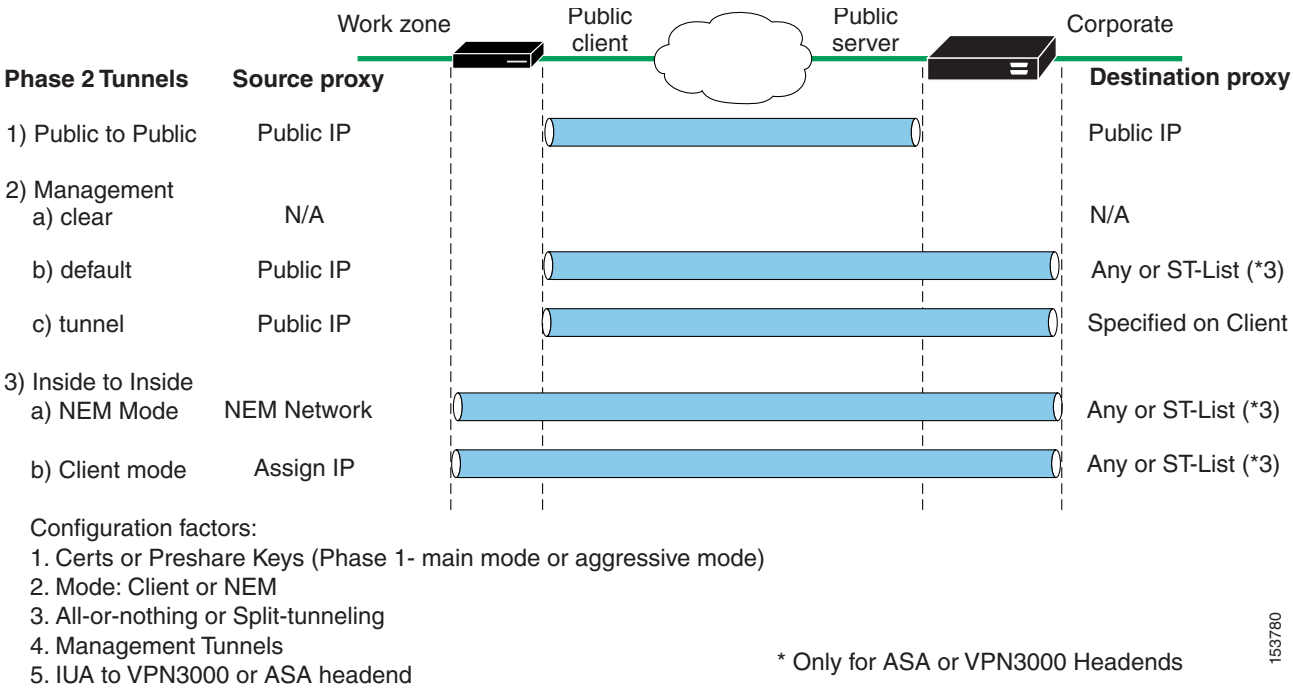
- Use of the **vpnclient management** command to specify one of the following automatic tunnel initiation options:
  - **tunnel** to limit administrative access to the client side by specific hosts or networks on the corporate side and use IPSec to add a layer of encryption to the management sessions over the HTTPS or SSH encryption that is already present.
  - **clear** to permit administrative access using the HTTPS or SSH encryption used by the management session.
  - **no** to prohibit management access

Caution Cisco does not support the use of the vpnclient management command if a NAT device is present between the client and the Internet.

- Use of the **vpnclient mode** command to specify one of the following modes of operation:
  - **client** to use Port Address Translation (PAT) mode to isolate the addresses of the inside hosts, relative to the client, from the enterprise network.
  - **network-extension-mode** to make those addresses accessible from the enterprise network.

Figure 36-1 shows the types of tunnels that the Easy VPN client initiates, based on the combination of the commands you enter.

Figure 36-1 Easy VPN Hardware Client Tunneling Options for the Cisco ASA 5505



The term "All-Or-Nothing" refers to the presence or absence of an access list for split tunneling. The access list ("ST-list") distinguishes networks that require tunneling from those that do not.

Specifying the Tunnel Group or Trustpoint

When configuring the Cisco ASA 5505 as an Easy VPN hardware client, you can specify a tunnel group or trustpoint configured on the Easy VPN server, depending on the Easy VPN server configuration. See the section that names the option you want to use:

- Specifying the Tunnel Group
- Specifying the Trustpoint

Specifying the Tunnel Group

Enter the following command in global configuration mode to specify the name of the VPN tunnel group and password for the Easy VPN client connection to the server:

vpnclient vpngroup *group_name* **password** *preshared_key*

group_name is the name of the VPN tunnel group configured on the Easy VPN server. You must configure this tunnel group on the server before establishing a connection.

preshared_key is the IKE pre-shared key used for authentication on the Easy VPN server.

For example, enter the following command to identify the VPN tunnel group named TestGroup1 and the IKE preshared key my_key123.

```
hostname(config)# vpnclient vpngroup TestGroup1 password my_key123
hostname(config)#
```

To remove the attribute from the running configuration, enter the following command:

no vpnclient vpngroup

If the configuration of the ASA 5505 running as an Easy VPN client does not specify a tunnel group, the client attempts to use an RSA certificate.

For example:

```
hostname(config)# no vpnclient vpngroup
hostname(config)#
```

Specifying the Trustpoint

A trustpoint represents a CA identity, and possibly a device identity, based on a certificate the CA issues. These parameters specify how the security appliance obtains its certificate from the CA and define the authentication policies for user certificates issued by the CA.

First define the trustpoint using the **crypto ca trustpoint** command, as described in "Configuring Trustpoints" section on page 41-7. Then enter the following command in global configuration mode to name the trustpoint identifying the RSA certificate to use for authentication:

vpnclient trustpoint *trustpoint_name* [**chain**]

trustpoint_name names the trustpoint identifying the RSA certificate to use for authentication.

(Optional) **chain** sends the entire certificate chain.

For example, enter the following command to specify the identity certificate named central and send the entire certificate chain:

```
hostname(config)# crypto ca trustpoint central
hostname(config)# vpnclient trustpoint central chain
hostname(config)#
```

To remove the attribute from the running configuration, enter the following command:

no vpnclient trustpoint

For example:

```
hostname(config)# no vpnclient trustpoint
hostname(config)#
```

Configuring Split Tunneling

Split tunneling lets a remote-access IPSec client conditionally direct packets over an IPSec tunnel in encrypted form or to a network interface in clear text form.

The Easy VPN server pushes the split tunneling attributes from the group policy to the Easy VPN Client for use only in the work zone. See Configuring Split-Tunneling Attributes, page 32-47 to configure split tunneling on the Cisco ASA 5505.

Enter the following command in global configuration mode to enable the automatic initiation of IPSec tunnels when NEM and split tunneling are configured:

[**no**] **vpnclient nem-st-autoconnect**

no removes the command from the running configuration.

For example:

```
hostname(config)# vpnclient nem-st-autoconnect
hostname(config)#
```

Configuring Device Pass-Through

Devices such as Cisco IP phones, wireless access points, and printers are incapable of performing authentication. Enter the following command in global configuration mode to exempt such devices from authentication, thereby providing network access to them, if individual user authentication is enabled:

[**no**] **vpnclient mac-exempt** *mac_addr_1 mac_mask_1* [*mac_addr_2 mac_mask_2...mac_addr_n mac_mask_n*]

no removes the command from the running configuration.

mac_addr is the MAC address, in dotted hexadecimal notation, of the device to bypass individual user authentication.

mac_mask is the network mask for the corresponding MAC address. A MAC mask of ffff.ff00.0000 matches all devices made by the same manufacturer. A MAC mask of ffff.ffff.ffff matches a single device.

Only the first six characters of the specific MAC address are required if you use the MAC mask ffff.ff00.0000 to specify all devices by the same manufacturer. For example, Cisco IP phones have the Manufacturer ID 00036b, so the following command exempts any Cisco IP phone, including Cisco IP phones, you might add in the future:

```
hostname(config)# vpnclient mac-exempt 0003.6b00.0000 ffff.ff00.0000
hostname(config)#
```

The next example provides greater security but less flexibility because it exempts one specific Cisco IP phone:

```
hostname(config)# vpnclient mac-exempt 0003.6b54.b213 ffff.ffff.ffff
hostname(config)#
```

Configuring Remote Management

The Cisco ASA 5505, operating as an Easy VPN hardware client, supports management access using SSH or HTTPS, with or without a second layer of additional encryption. You can configure the Cisco ASA 5505 to require IPSec encryption within the SSH or HTTPS encryption.

Use the **vpnclient management clear** command in global configuration mode to use normal routing to provide management access from the corporate network to the outside interface of the ASA 5505 (no tunneling management packets).

Caution Do not configure a management tunnel on a Cisco ASA 5505 configured as an Easy VPN hardware client if a NAT device is operating between the Easy VPN hardware client and the Internet. In that configuration, use the **vpnclient management clear** command.

Use the **vpnclient management tunnel** command in global configuration mode if you want to automate the creation of IPSec tunnels to provide management access from the corporate network to the outside interface of the ASA 5505. The Easy VPN hardware client and server create the tunnels automatically after the execution of the **vpnclient server** command. The syntax of the vpnclient management tunnel command follows:

vpnclient management tunnel *ip_addr_1 ip_mask_1* [*ip_addr_2 ip_mask_2...ip_addr_n ip_mask_n*]

For example, enter the following command to automate the creation of an IPSec tunnel to provide management access to the host with IP address 192.168.10.10:

```
hostname(config)# vpnclient management tunnel 192.198.10.10 255.255.255.0
hostname(config)#
```

The **no** form of this command sets up IPSec for management tunnels in accordance with the **split-tunnel-policy** and **split-tunnel-network-list** commands.

no vpnclient management

For example:

```
hostname(config)# no vpnclient management
hostname(config)#
```

Guidelines for Configuring the Easy VPN Server

The following sections address the Easy VPN hardware client considerations that apply to the Easy VPN server:

- Group Policy and User Attributes Pushed to the Client
- Authentication Options

Group Policy and User Attributes Pushed to the Client

Upon tunnel establishment, the Easy VPN server pushes the values of the group policy or user attributes stored in its configuration to the Easy VPN hardware client. Therefore, to change certain attributes pushed to the Easy VPN hardware client, you must modify them on the security appliances configured as the primary and secondary Easy VPN servers. This section identifies the group policy and user attributes pushed to the Easy VPN hardware client.

Note This section serves only as a reference. For complete instructions on configuring group policies and users, see Configuring Connection Profiles, Group Policies, and Users, page 32-1.

Use Table 36-2 as a guide for determining which commands to enter to modify the group policy or user attributes.

Table 36-2 Group Policy and User Attributes Pushed to the Cisco ASA 5505 Configured as an EasyVPN Hardware Client

Command	Description
backup-servers	Sets up backup servers on the client in case the primary server fails to respond.
banner	Sends a banner to the client after establishing a tunnel.
client-access-rule	Applies access rules.
client-firewall	Sets up the firewall parameters on the VPN client.
default-domain	Sends a domain name to the client.
dns-server	Specifies the IP address of the primary and secondary DNS servers, or prohibits the use of DNS servers.
dhcp-network-scope	Specifies the IP subnetwork to which the DHCP server assigns address to users within this group.
group-lock	Specifies a tunnel group to ensure that users connect to that group.
ipsec-udp	Uses UDP encapsulation for the IPSec tunnels.
ipsec-udp-port	Specifies the port number for IPSec over UDP.
nem	Enables or disables network extension mode.
password-storage	Lets the VPN user save a password in the user profile.
pfs	Commands the VPN client to use perfect forward secrecy.
re-xauth	Requires XAUTH authentication when IKE rekeys. **Note**: Disable re-xauth if secure unit authentication is enabled.
secure-unit-authentication	Enables interactive authentication for VPN hardware clients.
split-dns	Pushes a list of domains for name resolution.

Table 36-2 Group Policy and User Attributes Pushed to the Cisco ASA 5505 Configured as an EasyVPN Hardware Client (continued)

Command	Description
split-tunnel-network-list	Specifies one of the following: • No access list exists for split tunneling. All traffic travels across the tunnel. • Identifies the access list the security appliance uses to distinguish networks that require tunneling and those that do not. Split tunneling lets a remote-access IPSec client conditionally direct packets over an IPSec tunnel in encrypted form, or to a network interface in cleartext form. With split-tunneling enabled, packets not bound for destinations on the other side of the IPSec tunnel do not have to be encrypted, sent across the tunnel, decrypted, and then routed to a final destination.
split-tunnel-policy	Lets a remote-access IPSec client conditionally direct packets over an IPSec tunnel in encrypted form, or to a network interface in cleartext form. Options include the following: • split-tunnel-policy—Indicates that you are setting rules for tunneling traffic. • excludespecified—Defines a list of networks to which traffic goes in the clear. • tunnelall—Specifies that no traffic goes in the clear or to any other destination than the Easy VPN server. Remote users reach Internet networks through the corporate network and do not have access to local networks. • tunnelspecified—Tunnels all traffic from or to the specified networks. This option enables split tunneling. It lets you create a network list of addresses to tunnel. Data to all other addresses travels in the clear, and is routed by the remote user's internet service provider.
user-authentication	Enables individual user authentication for hardware-based VPN clients.
vpn-access-hours	Restricts VPN access hours.
vpn-filter	Applies a filter to VPN traffic.
vpn-idle-timeout	Specifies the number of minutes a session can be idle before it times out.
vpn-session-timeout	Specifies the maximum number of minutes for VPN connections.
vpn-simultaneous-logins	Specifies the maximum number of simultaneous logins.
vpn-tunnel-protocol	Specifies the permitted tunneling protocols.
wins-server	Specifies the IP address of the primary and secondary WINS servers, or prohibits the use of WINS servers.

Note IPSec NAT-T connections are the only IPSec connection types supported on the home VLAN of a Cisco ASA 5505. IPSec over TCP and native IPSec connections are not supported.

Authentication Options

The ASA 5505 supports the following authentication mechanisms, which it obtains from the group policy stored on the Easy VPN Server. The following list identifies the authentication options supported by the Easy VPN hardware client, however, you must configure them on the Easy VPN server:

- Secure unit authentication (SUA, also called Interactive unit authentication)

  Ignores the **vpnclient username** Xauth command (described in "Configuring Automatic Xauth Authentication" section on page 36-4) and requires the user to authenticate the ASA 5505 by entering a password. By default, SUA is disabled. You can use the **secure-unit-authentication enable** command in group-policy configuration mode to enable SUA. See Configuring Secure Unit Authentication, page 32-50.

- Individual user authentication

  Requires users behind the ASA 5505 to authenticate before granting them access to the enterprise VPN network. By default, IUA is disabled. To enable the IUA, use the **user-authentication enable** command in group-policy configuration mode. See Configuring User Authentication, page 32-51.

  The security appliance works correctly from behind a NAT device, and if the ASA5505 is configured in NAT mode, the provisioned IP (to which the clients all PAT) is injected into the routing table on the central-site device.

Caution Do not configure IUA on a Cisco ASA 5505 configured as an Easy VPN server if a NAT device is operating between the server and the Easy VPN hardware client.

  Use the **user-authentication-idle-timeout** command to set or remove the idle timeout period after which the Easy VPN Server terminates the client's access. See Configuring an Idle Timeout, page 32-51.

- Authentication by HTTP redirection

  The Cisco Easy VPN server intercepts HTTP traffic and redirects the user to a login page if one of the following is true:

  – SUA or the username and password are not configured on the Easy VPN hardware client.

  – IAU is enabled.

  HTTP redirection is automatic and does not require configuration on the Easy VPN Server.

- Preshared keys, digital certificates, tokens and no authentication

  The ASA 5505 supports preshared keys, token-based (e.g., SDI one-time passwords), and "no user authentication" for user authentication. **NOTE**: The Cisco Easy VPN server can use the digital certificate as part of user authorization. See Chapter 29, "Configuring IPsec and ISAKMP" for instructions.



CHAPTER 37

Configuring the PPPoE Client

This section describes how to configure the PPPoE client provided with the security appliance. It includes the following topics:

- PPPoE Client Overview, page 37-1
- Configuring the PPPoE Client Username and Password, page 37-2
- Enabling PPPoE, page 37-3
- Using PPPoE with a Fixed IP Address, page 37-3
- Monitoring and Debugging the PPPoE Client, page 37-4
- Using Related Commands, page 37-5

PPPoE Client Overview

PPPoE combines two widely accepted standards, Ethernet and PPP, to provide an authenticated method of assigning IP addresses to client systems. PPPoE clients are typically personal computers connected to an ISP over a remote broadband connection, such as DSL or cable service. ISPs deploy PPPoE because it supports high-speed broadband access using their existing remote access infrastructure and because it is easier for customers to use.

PPPoE provides a standard method of employing the authentication methods of the Point-to-Point Protocol (PPP) over an Ethernet network. When used by ISPs, PPPoE allows authenticated assignment of IP addresses. In this type of implementation, the PPPoE client and server are interconnected by Layer 2 bridging protocols running over a DSL or other broadband connection.

PPPoE is composed of two main phases:

- Active Discovery Phase—In this phase, the PPPoE client locates a PPPoE server, called an access concentrator. During this phase, a Session ID is assigned and the PPPoE layer is established.
- PPP Session Phase—In this phase, PPP options are negotiated and authentication is performed. Once the link setup is completed, PPPoE functions as a Layer 2 encapsulation method, allowing data to be transferred over the PPP link within PPPoE headers.

At system initialization, the PPPoE client establishes a session with the access concentrator by exchanging a series of packets. Once the session is established, a PPP link is set up, which includes authentication using Password Authentication protocol (PAP). Once the PPP session is established, each packet is encapsulated in the PPPoE and PPP headers.

Note PPPoE is not supported when failover is configured on the security appliance, or in multiple context or transparent mode. PPPoE is only supported in single, routed mode, without failover.

Configuring the PPPoE Client Username and Password

To configure the username and password used to authenticate the security appliance to the access concentrator, use the **vpdn** command. To use the **vpdn** command, you first define a VPDN group and then create individual users within the group.

To configure a PPPoE username and password, perform the following steps:

Step 1 Define the VPDN group to be used for PPPoE using the following command:

```
hostname(config)# vpdn group group_name request dialout pppoe
```

In this command, replace *group_name* with a descriptive name for the group, such as "pppoe-sbc."

Step 2 If your ISP requires authentication, select an authentication protocol by entering the following command:

```
hostname(config)# vpdn group group_name ppp authentication {chap | mschap | pap}
```

Replace *group_name* with the same group name you defined in the previous step. Enter the appropriate keyword for the type of authentication used by your ISP:

- CHAP—Challenge Handshake Authentication Protocol
- MS-CHAP—Microsoft Challenge Handshake Authentication Protocol Version 1
- PAP—Password Authentication Protocol

Note When using CHAP or MS-CHAP, the username may be referred to as the remote system name, while the password may be referred to as the CHAP secret.

Step 3 Associate the username assigned by your ISP to the VPDN group by entering the following command:

```
hostname(config)# vpdn group group_name localname username
```

Replace *group_name* with the VPDN group name and *username* with the username assigned by your ISP.

Step 4 Create a username and password pair for the PPPoE connection by entering the following command:

```
hostname(config)# vpdn username username password password [store-local]
```

Replace *username* with the username and *password* with the password assigned by your ISP.

Note The **store-local** option stores the username and password in a special location of NVRAM on the security appliance. If an Auto Update Server sends a **clear config** command to the security appliance and the connection is then interrupted, the security appliance can read the username and password from NVRAM and re-authenticate to the Access Concentrator.

Enabling PPPoE

Note You must complete the configuration using the **vpdn** command, described in "Configuring the PPPoE Client Username and Password," before enabling PPPoE.

The PPPoE client functionality is turned off by default. To enable PPPoE, perform the following steps:

Step 1 Enable the PPPoE client by entering the following command from interface configuration mode:

```
hostname(config-if)# ip address pppoe [setroute]
```

You can use a default route, or you can use the **setroute** option to set the default routes when the PPPoE client has not yet established a connection.

PPPoE is not supported in conjunction with DHCP because with PPPoE the IP address is assigned by PPP. The **setroute** option causes a default route to be created if no default route exists. The default router is the address of the access concentrator. The maximum transmission unit (MTU) size is automatically set to 1492 bytes, which is the correct value to allow PPPoE transmission within an Ethernet frame.

Reenter this command to reset the DHCP lease and request a new lease.

Note If PPPoE is enabled on two interfaces (such as a primary and backup interface), and you do not configure dual ISP support (see the "Configuring Static Route Tracking" section on page 10-5), then the security appliance can only send traffic through the first interface to acquire an IP address.

For example:

```
hostname(config)# interface gigabitethernet 0/0
hostname(config-if)# ip address pppoe
```

Step 2 Specify a VPDN group for the PPPoE client to use with the following command from interface configuration mode (optional):

```
hostname(config-if)# pppoe client vpdn group grpname
```

grpname is the name of a VPDN *group.*

Note If you have multiple VPDN groups configured, and you do not specify a group with the **pppoe client vpdn group** command, the security appliance may randomly choose a VPDN group. To avoid this, specify a VPDN group.

Using PPPoE with a Fixed IP Address

You can also enable PPPoE by manually entering the IP address, using the ip address command from interface configuration mode in the following format:

```
hostname(config-if)# ip address ipaddress mask pppoe
```

This command causes the security appliance to use the specified address instead of negotiating with the PPPoE server to assign an address dynamically. Replace *ipaddress* and *mask* with the IP address and subnet mask assigned to your security appliance.

For example:

```
hostname(config-if)# ip address outside 201.n.n.n 255.255.255.0 pppoe
```

Note The **setroute** option is an option of the **ip address** command that you can use to allow the access concentrator to set the default routes when the PPPoE client has not yet established a connection. When using the **setroute** option, you cannot have a statically defined route in the configuration.

Monitoring and Debugging the PPPoE Client

Use the following command to display the current PPPoE client configuration information:

```
hostname# show ip address outside pppoe
```

Use the following command to enable or disable debugging for the PPPoE client:

```
hostname# [no] debug pppoe {event | error | packet}
```

The following summarizes the function of each keyword:

- **event**—Displays protocol event information
- **error**—Displays error messages
- **packet**—Displays packet information

Use the following command to view the status of PPPoE sessions:

```
hostname# show vpdn session [l2tp | pppoe] [id sess_id | packets | state | window]
```

The following example shows a sample of information provided by this command:

```
hostname# show vpdn pppinterface

Tunnel id 0, 1 active sessions
     time since change 65862 secs
     Remote Internet Address 10.0.0.1
    Local Internet Address 199.99.99.3
     6 packets sent, 6 received, 84 bytes sent, 0 received
Remote Internet Address is 10.0.0.1
     Session state is SESSION_UP
       Time since event change 65865 secs, interface outside
       PPP interface id is 1
       6 packets sent, 6 received, 84 bytes sent, 0 received
hostname#
hostname# show vpdn session pppoe state
PPPoE Session Information (Total tunnels=1 sessions=1)
Remote Internet Address is 10.0.0.1
  Session state is SESSION_UP
    Time since event change 65887 secs, interface outside
    PPP interface id is 1
    6 packets sent, 6 received, 84 bytes sent, 0 received
hostname#
hostname# show vpdn tunnel pppoe state
PPPoE Tunnel Information (Total tunnels=1 sessions=1)
Tunnel id 0, 1 active sessions
   time since change 65901 secs
   Remote Internet Address 10.0.0.1
```

```
   Local Internet Address 199.99.99.3
   6 packets sent, 6 received, 84 bytes sent, 0 received
hostname#
```

Clearing the Configuration

To remove all **vpdn group** commands from the configuration, use the **clear configure vpdn group** command in global configuration mode:

```
hostname(config)# clear configure vpdn group
```

To remove all **vpdn username** commands, use the **clear configure vpdn username** command:

```
hostname(config)# clear configure vpdn username
```

Entering either of these commands has no affect upon active PPPoE connections.

Using Related Commands

Use the following command to cause the DHCP server to use the WINS and DNS addresses provided by the access concentrator as part of the PPP/IPCP negotiations:

```
hostname(config)# dhcpd auto_config [client_ifx_name]
```

This command is only required if the service provider provides this information as described in RFC 1877. The *client_ifx_name* parameter identifies the interface supported by the DHCP **auto_config** option. At this time, this keyword is not required because the PPPoE client is only supported on a single outside interface.



CHAPTER 38

Configuring LAN-to-LAN IPsec VPNs

A LAN-to-LAN VPN connects networks in different geographic locations.

Note The ASA supports LAN-to-LAN IPsec connections with Cisco peers, and with third-party peers that comply with all relevant standards.

This chapter describes how to build a LAN-to-LAN VPN connection. It includes the following sections:

- Summary of the Configuration, page 38-1
- Configuring Interfaces, page 38-2
- Configuring ISAKMP Policy and Enabling ISAKMP on the Outside Interface, page 38-2
- Creating a Transform Set, page 38-4
- Configuring an ACL, page 38-4
- Defining a Tunnel Group, page 38-5
- Creating a Crypto Map and Applying It To an Interface, page 38-6

Summary of the Configuration

This section provides a summary of the example LAN-to-LAN configuration this chapter creates. Later sections provide step-by-step instructions.

```
hostname(config)# interface ethernet0
hostname(config-if)# ip address 10.10.4.100 255.255.0.0
hostname(config-if)# no shutdown
hostname(config)# isakmp policy 1 authentication pre-share
hostname(config)# isakmp policy 1 encryption 3des
hostname(config)# isakmp policy 1 hash sha
hostname(config)# isakmp policy 1 group 2
hostname(config)# isakmp policy 1 lifetime 43200
hostname(config)# isakmp enable outside
hostname(config)# crypto ipsec transform set FirstSet esp-3des esp-md5-hmac
hostname(config)# access-list 121_list extended permit ip 192.168.0.0 255.255.0.0
150.150.0.0 255.255.0.0
hostname(config)# tunnel-group 10.10.4.108 type ipsec-121
hostname(config)# tunnel-group 10.10.4.108 ipsec-attributes
hostname(config-ipsec)# pre-shared-key 44kkaol59636jnfx
hostname(config)# crypto map abcmap 1 match address 121_list
hostname(config)# crypto map abcmap 1 set peer 10.10.4.108
hostname(config)# crypto map abcmap 1 set transform-set FirstSet
```

```
hostname(config)# crypto map abcmap interface outside
hostname(config)# write memory
```

Configuring Interfaces

A security appliance has at least two interfaces, referred to here as outside and inside. Typically, the outside interface is connected to the public Internet, while the inside interface is connected to a private network and is protected from public access.

To begin, configure and enable two interfaces on the security appliance. Then, assign a name, IP address and subnet mask. Optionally, configure its security level, speed, and duplex operation on the security appliance.

To configure interfaces, perform the following steps, using the command syntax in the examples:

Step 1 To enter Interface configuration mode, in global configuration mode enter the **interface** command with the default name of the interface to configure. In the following example the interface is ethernet0.

```
hostname(config)# interface ethernet0
hostname(config-if)#
```

Step 2 To set the IP address and subnet mask for the interface, enter the **ip address** command. In the following example the IP address is 10.10.4.100 and the subnet mask is 255.255.0.0.

```
hostname(config-if)# ip address 10.10.4.100 255.255.0.0
hostname(config-if)#
```

Step 3 To name the interface, enter the **nameif** command, maximum of 48 characters. You cannot change this name after you set it. In the following example the name of the ethernet0 interface is outside.

```
hostname(config-if)# nameif outside
hostname(config-if)#
```

Step 4 To enable the interface, enter the **no** version of the **shutdown** command. By default, interfaces are disabled.

```
hostname(config-if)# no shutdown
hostname(config-if)#
```

Step 5 To save your changes, enter the **write memory** command.

```
hostname(config-if)# write memory
hostname(config-if)#
```

Step 6 To configure a second interface, use the same procedure.

Configuring ISAKMP Policy and Enabling ISAKMP on the Outside Interface

The Internet Security Association and Key Management Protocol, also called IKE, is the negotiation protocol that lets two hosts agree on how to build an IPsec security association. Each ISAKMP negotiation is divided into two sections called Phase1 and Phase 2.

Phase 1 creates the first tunnel, which protects later ISAKMP negotiation messages. Phase 2 creates the tunnel that protects data travelling across the secure connection.

To set the terms of the ISAKMP negotiations, you create an ISAKMP policy, which includes the following:

- An authentication method, to ensure the identity of the peers.
- An encryption method, to protect the data and ensure privacy.
- A Hashed Message Authentication Codes method to ensure the identity of the sender and to ensure that the message has not been modified in transit.
- A Diffie-Hellman group to establish the strength of the encryption-key-determination algorithm. The security appliance uses this algorithm to derive the encryption and hash keys.
- A time limit for how long the security appliance uses an encryption key before replacing it.

See on page 29-3 in the "Configuring IPsec and ISAKMP" chapter of this guide for detailed information about the IKE policy keywords and their values.

To configure ISAKMP policies, in global configuration mode use the **isakmp policy** command with its various arguments. The syntax for ISAKMP policy commands is as follows:

isakmp policy *priority* **attribute_name** [**attribute_value** | *integer*].

Perform the following steps and use the command syntax in the following examples as a guide.

Step 1 Set the authentication method. The following example configures a preshared key. The priority is 1 in this and all following steps.

```
hostname(config)# isakmp policy 1 authentication pre-share
hostname(config)#
```

Step 2 Set the encryption method. The following example configures 3DES.

```
hostname(config)# isakmp policy 1 encryption 3des
hostname(config)#
```

Step 3 Set the HMAC method. The following example configures SHA-1.

```
hostname(config)# isakmp policy 1 hash sha
hostname(config)#
```

Step 4 Set the Diffie-Hellman group. The following example configures Group 2.

```
hostname(config)# isakmp policy 1 group 2
hostname(config)#
```

Step 5 Set the encryption key lifetime. The following example configures 43,200 seconds (12 hours).

```
hostname(config)# isakmp policy 1 lifetime 43200
hostname(config)#
```

Step 6 Enable ISAKMP on the interface named outside.

```
hostname(config)# isakmp enable outside
hostname(config)#
```

Step 7 To save your changes, enter the **write memory** command.

```
hostname(config)# write memory
hostname(config)#
```

Creating a Transform Set

A transform set combines an encryption method and an authentication method. During the IPsec security association negotiation with ISAKMP, the peers agree to use a particular transform set to protect a particular data flow. The transform set must be the same for both peers.

A transform set protects the data flows for the access list specified in the associated crypto map entry. You can create transform sets in the security appliance configuration, and then specify a maximum of 11 of them in a crypto map or dynamic crypto map entry.

Table 38-1 lists valid encryption and authentication methods.

Table 38-1 Valid Encryption and Authentication Methods

Valid Encryption Methods	Valid Authentication Methods
esp-des	esp-md5-hmac
esp-3des (default)	esp-sha-hmac (default)
esp-aes (128-bit encryption)	
esp-aes-192	
esp-aes-256	
esp-null	

Tunnel Mode is the usual way to implement IPsec between two security appliances that are connected over an untrusted network, such as the public Internet. Tunnel mode is the default and requires no configuration.

To configure a transform set, perform the following steps:

Step 1 In global configuration mode enter the **crypto ipsec transform-set** command. The following example configures a transform set with the name FirstSet, esp-3des encryption, and esp-md5-hmac authentication. The syntax is as follows:

crypto ipsec transform-set *transform-set-name* ***encryption-method authentication-method***

```
hostname(config)# crypto ipsec transform-set FirstSet esp-3des esp-md5-hmac
hostname(config)#
```

Step 2 Save your changes.

```
hostname(config)# write memory
hostname(config)#
```

Configuring an ACL

The security appliance uses access control lists to control network access. By default, the security appliance denies all traffic. You need to configure an ACL that permits traffic.

The ACLs that you configure for this LAN-to-LAN VPN control connections are based on the source and destination IP addresses. Configure ACLs that mirror each other on both sides of the connection.

To configure an ACL, perform the following steps:

Step 1 Enter the **access-list extended** command. The following example configures an ACL named l2l_list that lets traffic from IP addresses in the 192.168.0.0 network travel to the 150.150.0.0 network. The syntax is **access-list** *listname* **extended permit ip** *source-ipaddress source-netmask destination-ipaddress destination-netmask.*

```
hostname(config)# access-list l2l_list extended permit ip 192.168.0.0 255.255.0.0
150.150.0.0 255.255.0.0
hostname(config)#
```

Step 2 Configure an ACL for the security appliance on the other side of the connection that mirrors the ACL above. In the following example the prompt for the peer is hostname2.

```
hostname2(config)# access-list l2l_list extended permit ip 150.150.0.0 255.255.0.0
192.168.0.0 255.255.0.0
hostname(config)#
```

Defining a Tunnel Group

A tunnel group is a set of records that contain tunnel connection policies. You configure a tunnel group to identify AAA servers, specify connection parameters, and define a default group policy. The security appliance stores tunnel groups internally.

There are two default tunnel groups in the security appliance system: DefaultRAGroup, which is the default IPsec remote-access tunnel group, and DefaultL2Lgroup, which is the default IPsec LAN-to-LAN tunnel group. You can modify them but not delete them. You can also create one or more new tunnel groups to suit your environment. The security appliance uses these groups to configure default tunnel parameters for remote access and LAN-to-LAN tunnel groups when there is no specific tunnel group identified during tunnel negotiation.

To establish a basic LAN-to-LAN connection, you must set two attributes for a tunnel group:

- Set the connection type to IPsec LAN-to-LAN.
- Configure an authentication method, in the following example, preshared key.

Note To use VPNs, including tunnel groups, the ASA must be in single-routed mode. The commands to configure tunnel-group parameters do not appear in any other mode.

Step 1 To set the connection type to IPsec LAN-to-LAN, enter the **tunnel-group** command. The syntax is **tunnel-group** *name* **type** *type,* where *name* is the name you assign to the tunnel group, and *type* is the type of tunnel. The tunnel types as you enter them in the CLI are:

- **ipsec-ra** (IPsec remote access)
- **ipsec-l2l** (IPsec LAN to LAN)

In the following example the name of the tunnel group is the IP address of the LAN-to-LAN peer, 10.10.4.108.

```
hostname(config)# tunnel-group 10.10.4.108 type ipsec-l2l
hostname(config)#
```

Note LAN-to-LAN tunnel groups that have names that are not an IP address can be used only if the tunnel authentication method is Digital Certificates and/or the peer is configured to use Aggressive Mode.

Step 2 To set the authentication method to preshared key, enter the ipsec-attributes mode and then enter the **pre-shared-key** command to create the preshared key. You need to use the same preshared key on both security appliances for this LAN-to-LAN connection.

The key is an alphanumeric string of 1-128 characters. In the following example the preshared key is 44kkaol59636jnfx.

```
hostname(config)# tunnel-group 10.10.4.108 ipsec-attributes
hostname(config-tunnel-ipsec)# pre-shared-key 44kkaol59636jnfx
```

Step 3 Save your changes.

```
hostname(config)# write memory
hostname(config)#
```

Creating a Crypto Map and Applying It To an Interface

Crypto map entries pull together the various elements of IPsec security associations, including the following:

- Which traffic IPsec should protect, which you define in an access list.
- Where to send IPsec-protected traffic, by identifying the peer.
- What IPsec security applies to this traffic, which a transform set specifies.
- The local address for IPsec traffic, which you identify by applying the crypto map to an interface.

For IPsec to succeed, both peers must have crypto map entries with compatible configurations. For two crypto map entries to be compatible, they must, at a minimum, meet the following criteria:

- The crypto map entries must contain compatible crypto access lists (for example, mirror image access lists). If the responding peer uses dynamic crypto maps, the entries in the security appliance crypto access list must be “permitted” by the peer’s crypto access list.
- The crypto map entries each must identify the other peer (unless the responding peer is using a dynamic crypto map).
- The crypto map entries must have at least one transform set in common.

If you create more than one crypto map entry for a given interface, use the sequence number (seq-num) of each entry to rank it: the lower the seq-num, the higher the priority. At the interface that has the crypto map set, the security appliance evaluates traffic against the entries of higher priority maps first.

Create multiple crypto map entries for a given interface if either of the following conditions exist:

- Different peers handle different data flows.
- You want to apply different IPsec security to different types of traffic (to the same or separate peers), for example, if you want traffic between one set of subnets to be authenticated, and traffic between another set of subnets to be both authenticated and encrypted. In this case, define the different types of traffic in two separate access lists, and create a separate crypto map entry for each crypto access list.

To create a crypto map and apply it to the outside interface in global configuration mode, enter several of the **crypto map** commands. These commands use a variety of arguments, but the syntax for all of them begin with **crypto map** *map-name-seq-num*. In the following example the map-name is abcmap, the sequence number is 1.

Enter these commands in global configuration mode:

Step 1 To assign an access list to a crypto map entry, enter the **crypto map match address** command.

The syntax is **crypto map** *map-name seq-num* **match address** *aclname*. In the following example the map name is abcmap, the sequence number is 1, and the access list name is **l2l_list**.

```
hostname(config)# crypto map abcmap 1 match address l2l_list
hostname(config)#
```

Step 2 To identify the peer (s) for the IPsec connection, enter the **crypto map set peer** command.

The syntax is **crypto map** *map-name seq-num* **set peer** {*ip_address1* | *hostname1*}[... *ip_address10* | *hostname10*]. In the following example the peer name is 10.10.4.108.

```
hostname(config)# crypto map abcmap 1 set peer 10.10.4.108
hostname(config)#
```

Step 3 To specify a transform set for a crypto map entry, enter the **crypto map set transform-set** command.

The syntax is **crypto map** *map-name seq-num* **set transform-set** *transform-set-name*. In the following example the transform set name is FirstSet.

```
hostname(config)# crypto map abcmap 1 set transform-set FirstSet
hostname(config)#
```

Applying Crypto Maps to Interfaces

You must apply a crypto map set to each interface through which IPsec traffic travels. The security appliance supports IPsec on all interfaces. Applying the crypto map set to an interface instructs the security appliance to evaluate all interface traffic against the crypto map set and to use the specified policy during connection or security association negotiations.

Binding a crypto map to an interface also initializes the runtime data structures, such as the security association database and the security policy database. When you later modify a crypto map in any way, the security appliance automatically applies the changes to the running configuration. It drops any existing connections and reestablishes them after applying the new crypto map.

Step 1 To apply the configured crypto map to the outside interface, enter the **crypto map interface** command. The syntax is **crypto map** *map-name* **interface** *interface-name.*

```
hostname(config)# crypto map abcmap interface outside
hostname(config)#
```

Step 2 Save your changes.

```
hostname(config)# write memory
hostname(config)#
```



CHAPTER 39

Configuring Clientless SSL VPN

This chapter describes:

- Getting Started, page 39-1
- Creating and Applying Clientless SSL VPN Resources, page 39-21
- Configuring Connection Profile Attributes for Clientless SSL VPN, page 39-22
- Configuring Group Policy and User Attributes for Clientless SSL VPN, page 39-22
- Configuring Browser Access to Client-Server Plug-ins, page 39-24
- Configuring Application Access, page 39-30
- Configuring File Access, page 39-47
- Using Clientless SSL VPN with PDAs, page 39-50
- Using E-Mail over Clientless SSL VPN, page 39-50
- Optimizing Clientless SSL VPN Performance, page 39-52
- Clientless SSL VPN End User Setup, page 39-56
- Capturing Data, page 39-84

Getting Started

Note When the security appliance is configured for Clientless SSL VPN, you cannot enable security contexts (also called firewall multimode) or Active/Active stateful failover. Therefore, these features become unavailable.

Clientless SSL VPN lets users establish a secure, remote-access VPN tunnel to a security appliance using a web browser. Users do not need a software or hardware client.

Clientless SSL VPN provides secure and easy access to a broad range of web resources and web-enabled applications from almost any computer on the Internet. They include:

- Internal websites
- Web-enabled applications
- NT/Active Directory file shares
- E-mail proxies, including POP3S, IMAP4S, and SMTPS

- MS Outlook Web Access
- Application Access (that is, smart tunnel or port forwarding access to other TCP-based applications)

Note The security appliance does not support the Microsoft Outlook Exchange (MAPI) proxy. Neither the smart tunnel feature nor port forwarding supports MAPI. For Microsoft Outlook Exchange communication using the MAPI protocol, remote users must use AnyConnect.

Clientless SSL VPN uses Secure Sockets Layer Protocol and its successor, Transport Layer Security to provide the secure connection between remote users and specific, supported internal resources that you configure at a central site. The security appliance recognizes connections that need to be proxied, and the HTTP server interacts with the authentication subsystem to authenticate users.

The network administrator provides access to resources by users of clientless SSL VPN sessions on a group basis. Users have no direct access to resources on the internal network.

The following sections address getting started with the configuration of clientless SSL VPN access:

- Observing Clientless SSL VPN Security Precautions
- Understanding Features Not Supported in Clientless SSL VPN
- Using SSL to Access the Central Site
- Authenticating with Digital Certificates
- Enabling Cookies on Browsers for Clientless SSL VPN
- Managing Passwords
- Using Single Sign-on with Clientless SSL VPN
- Authenticating with Digital Certificates

Observing Clientless SSL VPN Security Precautions

Clientless SSL VPN connections on the security appliance differ from remote access IPSec connections, particularly with respect to how they interact with SSL-enabled servers, and precautions to follow to reduce security risks.

In a clientless SSL VPN connection, the security appliance acts as a proxy between the end user web browser and target web servers. When a user connects to an SSL-enabled web server, the security appliance establishes a secure connection and validates the server SSL certificate. The end user browser never receives the presented certificate, so therefore cannot examine and validate the certificate.

The current implementation of clientless SSL VPN on the security appliance does not permit communication with sites that present expired certificates. Nor does the security appliance perform trusted CA certificate validation. Therefore, users cannot analyze the certificate an SSL-enabled web-server presents before using with the web-enabled service.

We recommend that you do the following to minimize risks posed by Clientless SSL VPN access:

Step 1 Configure a group policy for all users who need Clientless SSL VPN access, and enable Clientless SSL VPN only for that group policy.

Step 2 Apply ACLs to permit access only to specific targets within the private network, permit access only to the private network, deny Internet access, or permit access only to reputable sites.

Step 3 Disable URL entry on the *portal page*, the page that opens when they establish a browser-based connection, to help prevent user access confusion. To do so, go to the **Configuration > Remote Access VPN > Clientless SSL VPN Access > Dynamic Access Policies** window, select the applicable DAP, click the **Edit > Functions** tab, and click **Disable** next to **URL entry**.

Step 4 Instruct users to enter external URLs in the native browser address field above the portal page or open a separate browser window to visit external sites.

Caution Users must not use the portal page (including the address field inside the portal page if you do not follow Step 3), to visit external sites that display the https prefix (for example, online banking sites).

Understanding Features Not Supported in Clientless SSL VPN

The security appliance does not support the following features for clientless SSL VPN connections:

- Inspection features under the Modular Policy Framework, inspecting configuration control.
- Functionality the filter configuration commands provide, including the **vpn-filter** command.
- VPN connections from hosts with IPv6 addresses. Hosts must use IPv4 addresses to establish Clientless SSL VPN or AnyConnect sessions. However, beginning with ASA 8.0(2), users can use these sessions to access internal IPv6-enabled resources.
- NAT, reducing the need for globally unique IP addresses.
- PAT, permitting multiple outbound sessions appear to originate from a single IP address.
- QoS, rate limiting using the **police** command and **priority-queue** command.
- Connection limits, checking either via the static or the Modular Policy Framework **set connection** command.
- The **established** command, allowing return connections from a lower security host to a higher security host if there is already an established connection from the higher level host to the lower level host.

Using SSL to Access the Central Site

Clientless SSL VPN uses SSL and its successor, TLS1 to provide a secure connection between remote users and specific, supported internal resources at a central site. This section includes the following topics:

- Using HTTPS for Clientless SSL VPN Sessions
- Configuring Clientless SSL VPN and ASDM Ports
- Configuring Support for Proxy Servers
- Configuring SSL/TLS Encryption Protocols

Using HTTPS for Clientless SSL VPN Sessions

Establishing clientless SSL VPN sessions requires the following:

- Enabling clientless SSL VPN sessions on the security appliance interface that users connect to.
- Using HTTPS to access the security appliance or load balancing cluster. In a web browser, users enter the security appliance IP address in the format *https:// address* where *address* is the IP address or DNS hostname of the security appliance interface.

To permit clientless SSL VPN sessions on an interface, perform the following steps:

Step 1 In global configuration mode, enter the **webvpn** command to enter webvpn mode.

Step 2 Enter the **enable** command with the name of the interface that you want to use for clientless SSL VPN sessions.

For example, to enable clientless SSL VPN sessions on the interface called outside, enter the following:

```
hostname(config)# webvpn
hostname(config-webvpn)# enable outside
```

Configuring Clientless SSL VPN and ASDM Ports

Beginning with Version 8.0(2), the security appliance supports both clientless SSL VPN sessions and ASDM administrative sessions simultaneously on Port 443 of the outside interface. You do, however, have the option to configure these applications on different interfaces.

To change the SSL listening port for clientless SSL VPN, use the **port** *port_number* command in webvpn mode. The following example enables clientless SSL VPN on port 444 of the outside interface. HTTPS for ASDM is also configured on the outside interface and uses the default port (443). With this configuration, remote users initiating clientless SSL VPN sessions enter https://<outside_ip>:444 in the browser.

```
hostname(config)# http server enable
hostname(config)# http 192.168.3.0 255.255.255.0 outside
hostname(config)# webvpn
hostname(config-webvpn)# port 444
hostname(config-webvpn)# enable outside
```

To change the listening port for ASDM, use the *port* argument of the **http server enable** command in privileged EXEC mode. The following example specifies that HTTPS ASDM sessions use port 444 on the outside interface. Clientless SSL VPN is also enabled on the outside interface and uses the default port (443). With this configuration, remote users initiate ASDM sessions by entering https://<outside_ip>:444 in the browser.

```
hostname(config)# http server enable 444
hostname(config)# http 192.168.3.0 255.255.255.0 outside
hostname(config)# webvpn
hostname(config-webvpn)# enable outside
```

Configuring Support for Proxy Servers

The security appliance can terminate HTTPS connections and forward HTTP and HTTPS requests to proxy servers. These servers act as intermediaries between users and the Internet. Requiring Internet access via a server that the organization controls provides another opportunity for filtering to assure secure Internet access and administrative control.

When configuring support for HTTP and HTTPS proxy services, you can assign preset credentials to send with each request for basic authentication. You can also specify URLs to exclude from HTTP and HTTPS requests.

You can specify a proxy autoconfiguration (PAC) file to download from an HTTP proxy server, however, you may not use proxy authentication when specifying the PAC file.

To configure the security appliance to use an external proxy server to handle HTTP and HTTPS requests, use the **http-proxy and https-proxy** commands in webvpn mode.

- **http-proxy** *host* [*port*] [**exclude** `url`] [**username** *username* {**password** *password*}]
- **https-proxy** *host* [*port*] [**exclude** `url`] [**username** *username* {**password** *password*}]
- **http-proxy pac** `url`

exclude—(Optional) Enter this keyword to exclude URLs from those that can be sent to the proxy server.

*host***—**Enter the hostname or IP address for the external proxy server.

pac—Proxy autoconfiguration file to download to the browser. Once downloaded, the PAC file uses a JavaScript function to identify a proxy for each URL.

password—(Optional, and available only if you specify a *username*) Enter this keyword to accompany each proxy request with a password to provide basic, proxy authentication.

*password***—**Enter the password to send to the proxy server with each HTTP or HTTPS request.

*port***—**(Optional) Enter the port number used by the proxy server. The default HTTP port is 80. The default HTTPS port is 443. The security appliance uses each of these ports if you do not specify an alternative value. The range is 1-65535.

`url`**—**If you entered **exclude**, enter a URL or a comma-delimited list of several URLs to exclude from those that can be sent to the proxy server. The string does not have a character limit, but the entire command cannot exceed 512 characters. You can specify literal URLs or use the following wildcards:

- ***** to match any string, including slashes (/) and periods (.). You must accompany this wildcard with an alphanumeric string.
- **?** to match any single character, including slashes and periods.
- [*x*-*y*] to match any single character in the range of *x* and *y*, where *x* represents one character and *y* represents another character in the ANSI character set.
- [**!***x*-*y*] to match any single character that is not in the range.

If you entered **http-proxy pac**, follow it with **http://** and type the URL of the proxy autoconfiguration file. If you omit the **http://** portion, the CLI ignores the command.

username—(Optional) Enter this keyword to accompany each HTTP proxy request with a username for basic, proxy authentication. Only the **http-proxy** *host* command supports this keyword.

*username***—**Enter the username the password to send to the proxy server with each HTTP or HTTPS request.

The security appliance clientless SSL VPN configuration supports only one **http-proxy** and one **http-proxy** command each. For example, if one instance of the **http-proxy** command is already present in the running configuration and you enter another, the CLI overwrites the previous instance.

The following example shows how to configure use of an HTTP proxy server with an IP address of 209.165. 201.1 using the default port, send a username and password with each HTTP request:

```
hostname(config-webvpn)# http-proxy 209.165.201.1 jsmith password mysecretdonttell
hostname(config-webvpn)
```

The following example shows the same command, except when the security appliance receives the specific URL www.example.com in an HTTP request, it resolves the request instead of passing it on to the proxy server:

```
hostname(config-webvpn)# http-proxy 209.165.201.1 exclude www.example.com username jsmith
password mysecretdonttell
hostname(config-webvpn)
```

The following example shows how to specify a URL to serve a proxy autoconfiguration file to the browser:

```
hostname(config-webvpn)# http-proxy pac http://www.example.com/pac
hostname(config-webvpn)
```

Configuring SSL/TLS Encryption Protocols

When you set SSL/TLS encryption protocols, be aware of the following:

- Make sure that the security appliance and the browser you use allow the same SSL/TLS encryption protocols.
- If you configure e-mail proxy, do not set the security appliance SSL version to TLSv1 Only. Microsoft Outlook and Microsoft Outlook Express do not support TLS.
- TCP Port Forwarding requires Sun Microsystems Java Runtime Environment (JRE) version 1.4.x and 1.5.x. Port forwarding does not work when a user of clientless SSL VPN connects with some SSL versions, as follows:

Negotiate SSLv3	Java downloads
Negotiate SSLv3/TLSv1	Java downloads
Negotiate TLSv1	Java does NOT download
TLSv1Only	Java does NOT download
SSLv3Only	Java does NOT download

Authenticating with Digital Certificates

SSL uses digital certificates for authentication. The security appliance creates a self-signed SSL server certificate when it boots; or you can install in the security appliance an SSL certificate that has been issued in a PKI context. For HTTPS, this certificate must then be installed on the client. You need to install the certificate from a given security appliance only once.

Restrictions for authenticating users with digital certificates include the following:

- Application Access does not work for users of clientless SSL VPN who authenticate using digital certificates. JRE does not have the ability to access the web browser keystore. Therefore JAVA cannot use a certificate that the browser uses to authenticate a user, so it cannot start.
- E-mail clients such as MS Outlook, MS Outlook Express, and Eudora lack the ability to access the certificate store.

For more information on authentication and authorization using digital certificates, see "Using Certificates and User Login Credentials" in the "Configuring AAA Servers and the Local Database" chapter.

Enabling Cookies on Browsers for Clientless SSL VPN

Browser cookies are required for the proper operation of clientless SSL VPN. When cookies are disabled on the web browser, the links from the web portal home page open a new window prompting the user to log in once more.

Managing Passwords

Optionally, you can configure the security appliance to warn end users when their passwords are about to expire. To do this, you specify the **password-management** command in tunnel-group general-attributes mode or enable the feature using ASDM at Configuration > Remote Access VPN > Clientless SSL VPN Access > Connection Profiles > Add or Edit > Advanced > General > Password Management.

The security appliance supports password management for the RADIUS and LDAP protocols. It supports the "password-expire-in-days" option for LDAP only.

You can configure password management for IPSec remote access and SSL VPN tunnel-groups.

When you configure password management, the security appliance notifies the remote user at login that the user's current password is about to expire or has expired. The security appliance then offers the user the opportunity to change the password. If the current password has not yet expired, the user can still log in using that password.

This command is valid for AAA servers that support such notification. The security appliance ignores this command if RADIUS or LDAP authentication has not been configured.



Note Some RADIUS servers that support MSCHAP currently do not support MSCHAPv2. This command requires MSCHAPv2 so please check with your vendor.

The security appliance, releases 7.1 and later, generally supports password management for the following connection types when authenticating with LDAP or with any RADIUS configuration that supports MS-CHAPv2:

- AnyConnect VPN Client
- IPSec VPN Client
- Clientless SSL VPN

Password management is *not* supported for any of these connection types for Kerberos/Active Directory (Windows password) or NT 4.0 Domain.

The RADIUS server (for example, Cisco ACS) could proxy the authentication request to another authentication server. However, from the security appliance perspective, it is talking only to a RADIUS server.



Note For LDAP, the method to change a password is proprietary for the different LDAP servers on the market. Currently, the security appliance implements the proprietary password management logic only for Microsoft Active Directory and Sun LDAP servers.

Native LDAP requires an SSL connection. You must enable LDAP over SSL before attempting to do password management for LDAP. By default, LDAP uses port 636.

Note If you are using an LDAP directory server for authentication, password management is supported with the Sun Microsystems JAVA System Directory Server (formerly named the Sun ONE Directory Server) and the Microsoft Active Directory.

Sun—The DN configured on the security appliance to access a Sun directory server must be able to access the default password policy on that server. We recommend using the directory administrator, or a user with directory administrator privileges, as the DN. Alternatively, you can place an ACI on the default password policy.

Microsoft—You must configure LDAP over SSL to enable password management with Microsoft Active Directory.

Note that this command does not change the number of days before the password expires, but rather, the number of days ahead of expiration that the security appliance starts warning the user that the password is about to expire.

If you do specify the **password-expire-in-days** keyword, you must also specify the number of days.

Specifying this command with the number of days set to 0 disables this command. The security appliance does not notify the user of the pending expiration, but the user can change the password after it expires.

The following example sets the days before password expiration to begin warning the user of the pending expiration to 90 for the connection profile “testgroup”:

```
hostname(config)# tunnel-group testgroup type webvpn
hostname(config)# tunnel-group testgroup general-attributes
hostname(config-general)# password-management password-expire-in-days 90
```

Using Single Sign-on with Clientless SSL VPN

Single sign-on support lets users of clientless SSL VPN enter a username and password only once to access multiple protected services and web servers. In general, the SSO mechanism either starts as part of the AAA process or just after successful user authentication to a AAA server. The clientless SSL VPN server running on the security appliance acts as a proxy for the user to the authenticating server. When a user logs in, the clientless SSL VPN server sends an SSO authentication request, including username and password, to the authenticating server using HTTPS. If the server approves the authentication request, it returns an SSO authentication cookie to the clientless SSL VPN server. The security appliance keeps this cookie on behalf of the user and uses it to authenticate the user to secure websites within the domain protected by the SSO server.

This section describes the three SSO authentication methods supported by clientless SSL VPN: HTTP Basic and NTLMv1 (NT LAN Manager) authentication, the Computer Associates eTrust SiteMinder SSO server (formerly Netegrity SiteMinder), and Version 1.1 of Security Assertion Markup Language (SAML), the POST-type SSO server authentication.

This section includes:

- Configuring SSO with HTTP Basic or NTLM Authentication
- Configuring SSO Authentication Using SiteMinder
- Configuring SSO Authentication Using SAML Browser Post Profile
- Configuring SSO with the HTTP Form Protocol

Configuring SSO with HTTP Basic or NTLM Authentication

This section describes single sign-on with HTTP Basic or NTLM authentication. You can configure the security appliance to implement SSO using either or both of these methods. The **auto-signon** command configures the security appliance to automatically pass clientless SSL VPN user login credentials (username and password) on to internal servers. You can enter multiple **auto-signon** commands. The security appliance processes them according to the input order (early commands take precedence). You specify the servers to receive the login credentials using either IP address and IP mask, or URI mask.

Use the **auto-signon** command in any of three modes: webvpn configuration, webvpn group-policy mode, or webvpn username mode. Username supersedes group, and group supersedes global. The mode you choose depends upon scope of authentication you want:

Mode	Scope
webvpn configuration	All clientless SSL VPN users globally
webvpn group-policy configuration	A subset of clientless SSL VPN users defined by a group policy
webvpn username configuration	An individual user of clientless SSL VPN

The following example commands present various possible combinations of modes and arguments.

All Users, IP Address Range, NTLM

To configure **auto-signon** for all users of clientless SSL VPN to servers with IP addresses ranging from 10.1.1.0 to 10.1.1.255 using NTLM authentication, for example, enter the following commands:

```
hostname(config)# webvpn
hostname(config-webvpn)# auto-signon allow ip 10.1.1.1 255.255.255.0 auth-type ntlm
```

All Users, URI Range, HTTP Basic

To configure **auto-signon** for all users of clientless SSL VPN, using basic HTTP authentication, to servers defined by the URI mask https://*.example.com/*, for example, enter the following commands:

```
hostname(config)# webvpn
hostname(config-webvpn)# auto-signon allow uri https://*.example.com/* auth-type basic
```

Group, URI Range, HTTP Basic and NTLM

To configure **auto-signon** for clientless SSL VPN sessions associated with the ExamplePolicy group policy, using either basic or NTLM authentication, to servers defined by the URI mask https://*.example.com/*, for example, enter the following commands:

```
hostname(config)# group-policy ExamplePolicy attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# auto-signon allow uri https://*.example.com/* auth-type all
```

Specific User, IP Address Range, HTTP Basic

To configure **auto-signon** for a user named Anyuser to servers with IP addresses ranging from 10.1.1.0 to 10.1.1.255 using HTTP Basic authentication, for example, enter the following commands:

```
hostname(config)# username Anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# auto-signon allow ip 10.1.1.1 255.255.255.0 auth-type
basic
```

Configuring SSO Authentication Using SiteMinder

This section describes configuring the security appliance to support SSO with SiteMinder. You would typically choose to implement SSO with SiteMinder if your website security infrastucture already incorporates SiteMinder. With this method, SSO authentication is separate from AAA and happens once the AAA process completes. If you want to configure SSO for a user or group for clientless SSL VPN access, you must first configure a AAA server, such as a RADIUS or LDAP server. You can then set up SSO support for clientless SSL VPN. This section includes:

- Task Overview: Configuring SSO with SiteMinder
- Detailed Tasks: Configuring SSO with SiteMinder
- Adding the Cisco Authentication Scheme to SiteMinder

Task Overview: Configuring SSO with SiteMinder

This section presents an overview of the tasks necessary to configure SSO with SiteMinder SSO. These tasks are:

- Specifying the SSO server.
- Specifying the URL of the SSO server to which the security appliance makes SSO authentication requests.
- Specifying a secret key to secure the communication between the security appliance and the SSO server. This key is similar to a password: you create it, save it, and enter it on both the security appliance and the SiteMinder Policy Server using the Cisco Java plug-in authentication scheme.

Optionally, you can do the following configuration tasks in addition to the required tasks:

- Configuring the authentication request timeout.
- Configuring the number of authentication request retries.

After you complete these tasks, assign an SSO server to a user or group policy.

Detailed Tasks: Configuring SSO with SiteMinder

This section presents specific steps for configuring the security appliance to support SSO authentication with CA SiteMinder. To configure SSO with SiteMinder, perform the following steps:

Step 1 In webvpn configuration mode, enter the **sso-server** command with the **type** option to create an SSO server. For example, to create an SSO server named Example of type siteminder, enter the following:

```
hostname(config)# webvpn
hostname(config-webvpn)# sso-server Example type siteminder
hostname(config-webvpn-sso-siteminder)#
```

Step 2 Enter the **web-agent-url** command in webvpn-sso-siteminder configuration mode to specify the authentication URL of the SSO server. For example, to send authentication requests to the URL http://www.Example.com/webvpn, enter the following:

```
hostname(config-webvpn-sso-siteminder)# web-agent-url http://www.Example.com/webvpn
hostname(config-webvpn-sso-siteminder)#
```

Step 3 Specify a secret key to secure the authentication communications between the security appliance and SiteMinder using the **policy-server-secret** command in webvpn-sso-siteminder configuration mode. You can create a key of any length using any regular or shifted alphanumeric character, but you must enter the same key on both the security appliance and the SSO server.

For example, to create the secret key AtaL8rD8!, enter the following:

```
hostname(config-webvpn-sso-siteminder)# policy-server-secret AtaL8rD8!
hostname(config-webvpn-sso-siteminder)#
```

Step 4 Optionally, you can configure the number of seconds before a failed SSO authentication attempt times out using the **request-timeout** command in webvpn-sso-siteminder configuration mode. The default number of seconds is 5 seconds and the possible range is 1 to 30 seconds. To change the number of seconds before a request times out to 8, for example, enter the following:

```
hostname(config-webvpn-sso-siteminder)# request-timeout 8
hostname(config-webvpn-sso-siteminder)#
```

Step 5 Optionally, you can configure the number of times the security appliance retries a failed SSO authentication attempt before the authentication times-out using the **max-retry-attempts** command in webvpn-sso-siteminder configuration mode. The default is 3 retry attempts and the possible range is 1 to 5 attempts. To configure the number of retries to be 4, for example, enter the following:

```
hostname(config-webvpn-sso-siteminder)# max-retry-attempts 4
hostname(config-webvpn-sso-siteminder)#
```

Step 6 After you configure the SSO server, you must specify SSO authentication for either a group or user. To specify SSO for a group, assign an SSO server to a group policy using the **sso-server value** command in group-policy-webvpn configuration mode. To specify SSO for a user, assign an SSO server to a user policy using the same command, **sso-server value**, but in username-webvpn configuration mode. For example, to assign the SSO server named Example to the user named Anyuser, enter the following:

```
hostname(config)# username Anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# sso-server value Example
hostname(config-username-webvpn)#
```

Step 7 Finally, you can test the SSO server configuration using the **test sso-server** command in privileged EXEC mode. For example, to test the SSO server named Example using the username Anyuser, enter the following:

```
hostname# test sso-server Example username Anyuser
INFO: Attempting authentication request to sso-server Example for user Anyuser
INFO: STATUS: Success
hostname#
```

Adding the Cisco Authentication Scheme to SiteMinder

In addition to configuring the security appliance for SSO with SiteMinder, you must also configure your CA SiteMinder Policy Server with the Cisco authentication scheme, a Java plug-in you download from the Cisco web site.



Note Configuring the SiteMinder Policy Server requires experience with SiteMinder. This section presents general tasks, not a complete procedure.

To configure the Cisco authentication scheme on your SiteMinder Policy Server, perform the following tasks:

Step 1 With the SiteMinder Administration utility, create a custom authentication scheme, being sure to use the following specific arguments:

- In the Library field, enter **smjavaapi**.
- In the Secret field, enter the same secret configured on the security appliance.

  You configure the secret on the security appliance using the **policy-server-secret** command at the command line interface.
- In the Parameter field, enter **CiscoAuthApi**.

Step 2 Using your Cisco.com login, download the file **cisco_vpn_auth.jar** from http://www.cisco.com/cgi-bin/tablebuild.pl/asa and copy it to the default library directory for the SiteMinder server. This .jar file is also available on the Cisco security appliance CD.

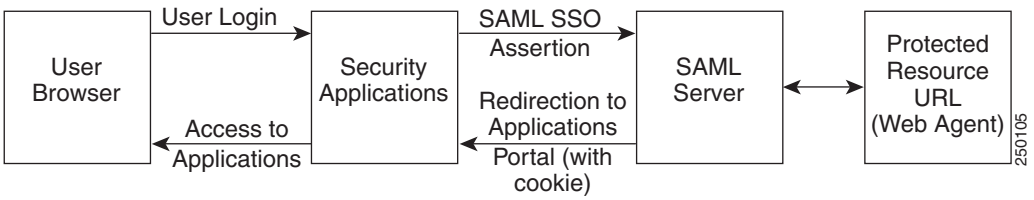
Configuring SSO Authentication Using SAML Browser Post Profile

This section describes configuring the security appliance to support Security Assertion Markup Language (SAML), Version 1.1 POST profile Single Sign-On (SSO) for authorized users. SAML SSO is supported only for clientless SSL VPN sessions. This section includes:

- Task Overview: Configuring SSO with SAML Post Profile
- Detailed Tasks: Configuring SSO with SAML Post Profile
- SSO Server Configuration

After a session is initiated, the security appliance authenticates the user against a configured AAA method. Next, the security appliance (the asserting party) generates an assertion to the relying party, the consumer URL service provided by the SAML server. If the SAML exchange succeeds, the user is allowed access to the protected resource. Figure 39-1 shows the communication flow:

Figure 39-1 SAML Communication Flow



Note The SAML Browser Artifact method of exchanging assertions is not supported.

Task Overview: Configuring SSO with SAML Post Profile

This section presents an overview of the tasks necessary to configure SSO with SAML Browser Post Profile. These tasks are:

- Specify the SSO server with the **sso-server** command.
- Specify the URL of the SSO server for authentication requests (the **assertion-consumer-url** command)

- Specify the security appliance hostname as the component issuing the authentication request (the **issuer** command)
- Specify the trustpoint certificates use for signing SAML Post Profile assertions (the **trustpoint** command)

Optionally, in addition to these required tasks, you can do the following configuration tasks:

- Configure the authentication request timeout (the **request-timeout** command)
- Configure the number of authentication request retries (the **max-retry-attempts** command)

After completing the configuration tasks, you assign an SSO server to a user or group policy.

Detailed Tasks: Configuring SSO with SAML Post Profile

This section presents specific steps for configuring the security appliance to support SSO authentication with SAML Post Profile. To configure SSO with SAML-V1.1-POST, perform the following steps:

Step 1 In webvpn configuration mode, enter the **sso-server** command with the **type** option to create an SSO server. For example, to create an SSO server named Sample of type SAML-V1.1-POST, enter the following:

```
hostname(config)# webvpn
hostname(config-webvpn)# sso-server sample type SAML-V1.1-post
hostname(config-webvpn-sso-saml)#
```



Note The security appliance currently supports only the Browser Post Profile type of SAML SSO Server.

Step 2 Enter the **assertion-consumer-url** command in webvpn-sso-saml configuration mode to specify the authentication URL of the SSO server. For example, to send authentication requests to the URL http://www.Example.com/webvpn, enter the following:

```
hostname(config-webvpn-sso-saml)# assertion-consumer-url http://www.sample.com/webvpn
hostname(config-webvpn-sso-saml)#
```

Step 3 Specify a unique string that identifies the security appliance itself when it generates assertions. Typically, this issuer name is the hostname for the security appliance as follows:

```
hostname(config-webvpn-sso-saml)# issuer myasa
hostname(config-webvpn-sso-saml)#
```

Step 4 Specify the identification certificate for signing the assertion with the **trust-point** command. An example follows:

```
hostname(config)# tunnel-group 209.165.200.225 type IPSec_L2L
hostname(config)# tunnel-group 209.165.200.225 ipsec-attributes
hostname(config-tunnel-ipsec)# trust-point mytrustpoint
```

Optionally, you can configure the number of seconds before a failed SSO authentication attempt times out using the **request-timeout** command in webvpn-sso-saml configuration mode. The default number of seconds is 5 seconds and the possible range is 1 to 30 seconds. To change the number of seconds before a request times out to 8, for example, enter the following:

```
hostname(config-webvpn-sso-saml)# request-timeout 8
hostname(config-webvpn-sso-saml)#
```

Step 5 Optionally, you can configure the number of times the security appliance retries a failed SSO authentication attempt before the authentication times-out using the **max-retry-attempts** command in webvpn-sso-saml configuration mode. The default is 3 retry attempts and the possible range is 1 to 5 attempts. To configure the number of retries to be 4, for example, enter the following:

```
hostname(config-webvpn-sso-saml)# max-retry-attempts 4
hostname(config-webvpn-sso-saml)#
```

Step 6 After you configure the SSO server, you must specify SSO authentication for either a group or user. To specify SSO for a group, assign an SSO server to a group policy using the **sso-server value** command in group-policy-webvpn configuration mode. To specify SSO for a user, assign an SSO server to a user policy using the same command, **sso-server value**, but in username-webvpn configuration mode. For example, to assign the SSO server named Example to the user named Anyuser, enter the following:

```
hostname(config)# username Anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# sso-server value sample
hostname(config-username-webvpn)#
```

Step 7 Finally, you can test the SSO server configuration using the **test sso-server** command in privileged EXEC mode. For example, to test the SSO server, Example using the username Anyuser, enter:

```
hostname# test sso-server Example username Anyuser
INFO: Attempting authentication request to sso-server sample for user Anyuser
INFO: STATUS: Success
```

SSO Server Configuration

Use the SAML server documentation provided by the server software vendor to configure the SAML server in Relying Party mode.The following steps list the specific parameters required to configure the SAML Server for Browser Post Profile:

Step 1 Configure the SAML server parameters to represent the asserting party (the security appliance):

- Recipient consumer url (same as the assertion consumer url configured on the ASA)
- Issuer ID, a string, usually the hostname of appliance
- Profile type -Browser Post Profile

Step 2 Configure certificates.

Step 3 Specify that asserting party assertions must be signed.

Step 4 Select how the SAML server identifies the user:

- Subject Name Type is DN
- Subject Name format is uid=<user>

Configuring SSO with the HTTP Form Protocol

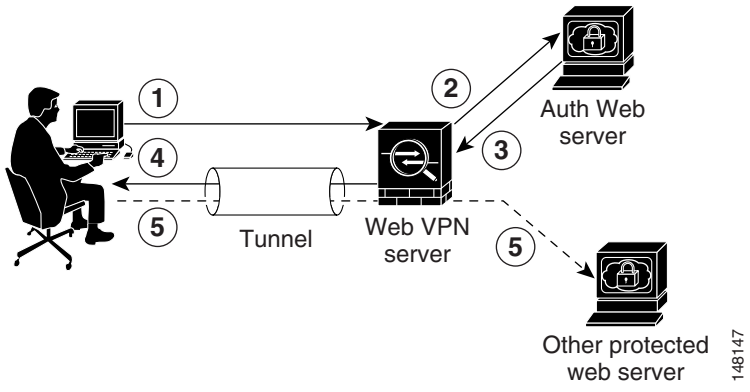
This section describes using the HTTP Form protocol for SSO. HTTP Form protocol is a common approach to SSO authentication that can also qualify as a AAA method. It provides a secure method for exchanging authentication information between users of clientless SSL VPN and authenticating web servers. As a common protocol, it is highly compatible with web servers and web-based SSO products, and you can use it in conjunction with other AAA servers such as RADIUS or LDAP servers.

Note To configure SSO with the HTTP protocol correctly, you must have a thorough working knowledge of authentication and HTTP protocol exchanges.

The security appliance again serves as a proxy for users of clientless SSL VPN to an authenticating web server but, in this case, it uses HTTP Form protocol and the POST method for requests. You must configure the security appliance to send and receive form data. Figure 39-2 illustrates the following SSO authentication steps:

1. A user of clientless SSL VPN first enters a username and password to log into the clientless SSL VPN server on the security appliance.
2. The clientless SSL VPN server acts as a proxy for the user and forwards the form data (username and password) to an authenticating web server using a POST authentication request.
3. If the authenticating web server approves the user data, it returns an authentication cookie to the clientless SSL VPN server where it is stored on behalf of the user.
4. The clientless SSL VPN server establishes a tunnel to the user.
5. The user can now access other websites within the protected SSO environment without reentering a username and password.

Figure 39-2 SSO Authentication Using HTTP Forms



While you would expect to configure form parameters that let the security appliance include POST data such as the username and password, you initially might not be aware of additional hidden parameters that the web server requires. Some authentication applications expect hidden data which is neither visible to nor entered by the user. You can, however, discover hidden parameters the authenticating web server expects by making a direct authentication request to the web server from your browser without the security appliance in the middle acting as a proxy. Analyzing the web server response using an HTTP header analyzer reveals hidden parameters in a format similar to the following:

```
<param name>=<URL encoded value>&<param name>=<URL encoded>
```

Some hidden parameters are mandatory and some are optional. If the web server requires data for a hidden parameter, it rejects any authentication POST request that omits that data. Because a header analyzer does not tell you if a hidden parameter is mandatory or not, we recommend that you include all hidden parameters until you determine which are mandatory.

This section describes:

- Gathering HTTP Form Data
- Task Overview: Configuring SSO with HTTP Form Protocol
- Detailed Tasks: Configuring SSO with HTTP Form Protocol

Gathering HTTP Form Data

This section presents the steps for discovering and gathering necessary HTTP Form data. If you do not know what parameters the authenticating web server requires, you can gather parameter data by analyzing an authentication exchange using the following steps:

Note These steps require a browser and an HTTP header analyzer.

Step 1 Start your browser and HTTP header analyzer, and connect directly to the web server login page without going through the security appliance.

Step 2 After the web server login page has loaded in your browser, examine the login sequence to determine if a cookie is being set during the exchange. If the web server has loaded a cookie with the login page, configure this login page URL as the *start-URL*.

Step 3 Enter the username and password to log in to the web server, and press Enter. This action generates the authentication POST request that you examine using the HTTP header analyzer.

An example POST request—with host HTTP header and body—follows:

```
POST
/emco/myemco/authc/forms/MCOlogin.fcc?TYPE=33554433&REALMOID=06-000430e1-7443-125c-ac05
-83846dc90034&GUID=&SMAUTHREASON=0&METHOD=GET&SMAGENTNAME=$SM$5FZmjnk3DRNwNjk2KcqVCFbIr
NT9%2bJ0H0KPshFtg6rB1UV2PxkHqLw%3d%3d&TARGET=https%3A%2F%2Fwww.example.com%2Femco%2Fmye
mco%2FHTTP/1.1

Host: www.example.com

(BODY)

SMENC=ISO-8859-1&SMLOCALE=US-EN&USERID=Anyuser&USER_PASSWORD=XXXXXX&target=https%3A%2F%
2Fwww.example.com%2Femco%2Fmyemco%2F&smauthreason=0
```

Step 4 Examine the POST request and copy the protocol, host, and the complete URL to configure the action-uri parameter.

Step 5 Examine the POST request body and copy the following:

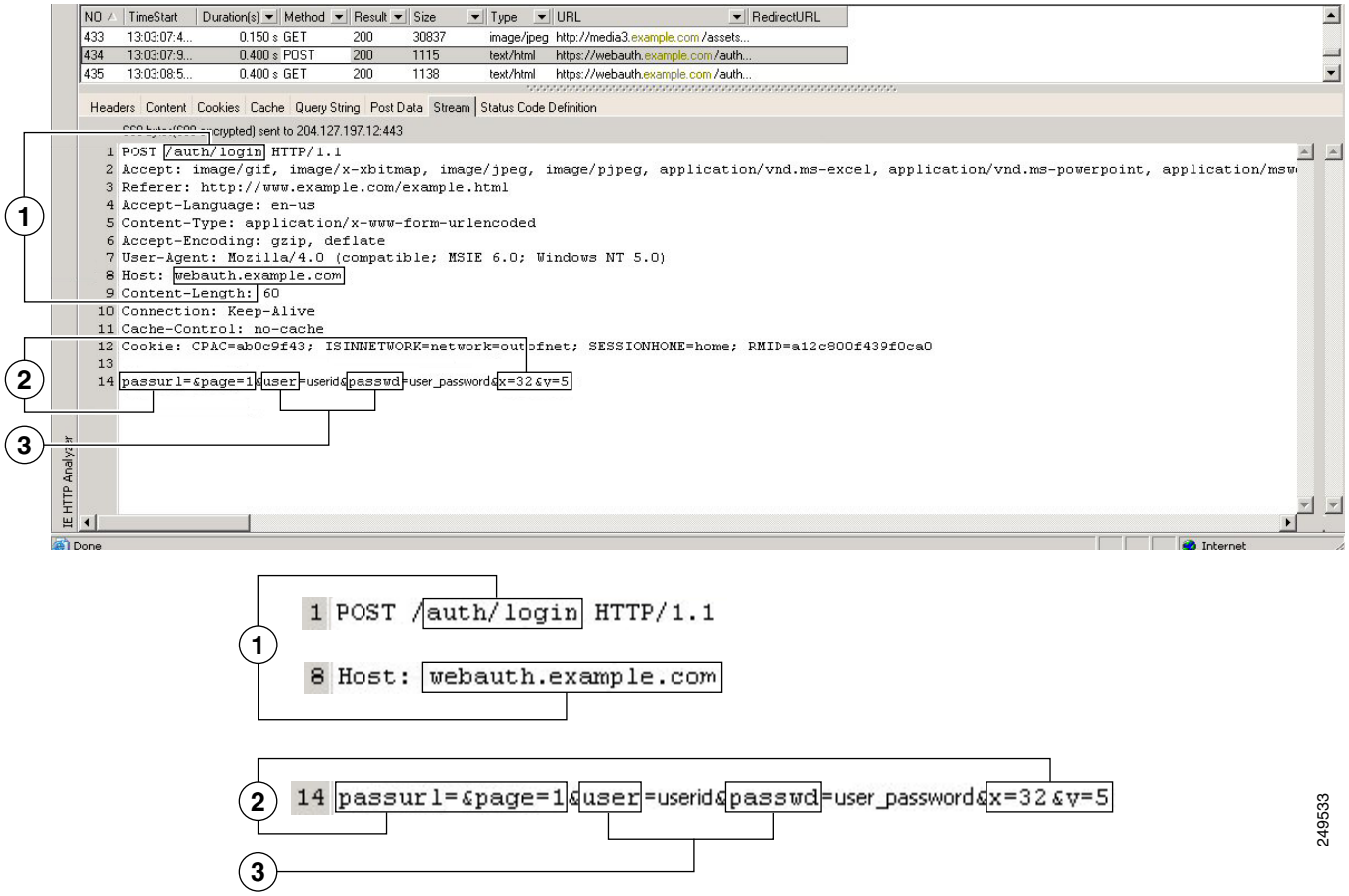
a. Username parameter. In the preceding example, this parameter is USERID, not the value anyuser.

b. Password parameter. In the preceding example, this parameter is USER_PASSWORD.

c. Hidden parameter. This parameter is everything in the POST body except the username and password parameters. In the preceding example, the hidden parameter is: SMENC=ISO-8859-1&SMLOCALE=US-EN&target=https%3A%2F%2Fwww.example.com%2Femco%2Fmyemco%2F&smauthreason=0

Figure 39-3 highlights the action URI, hidden, username and password parameters within sample output from an HTTP analyzer. This is only an example; output varies widely across different websites.

Figure 39-3 Action-uri, hidden, username and password parameters



1	Action URI parameter
2	Hidden parameters
3	Username and password parameters

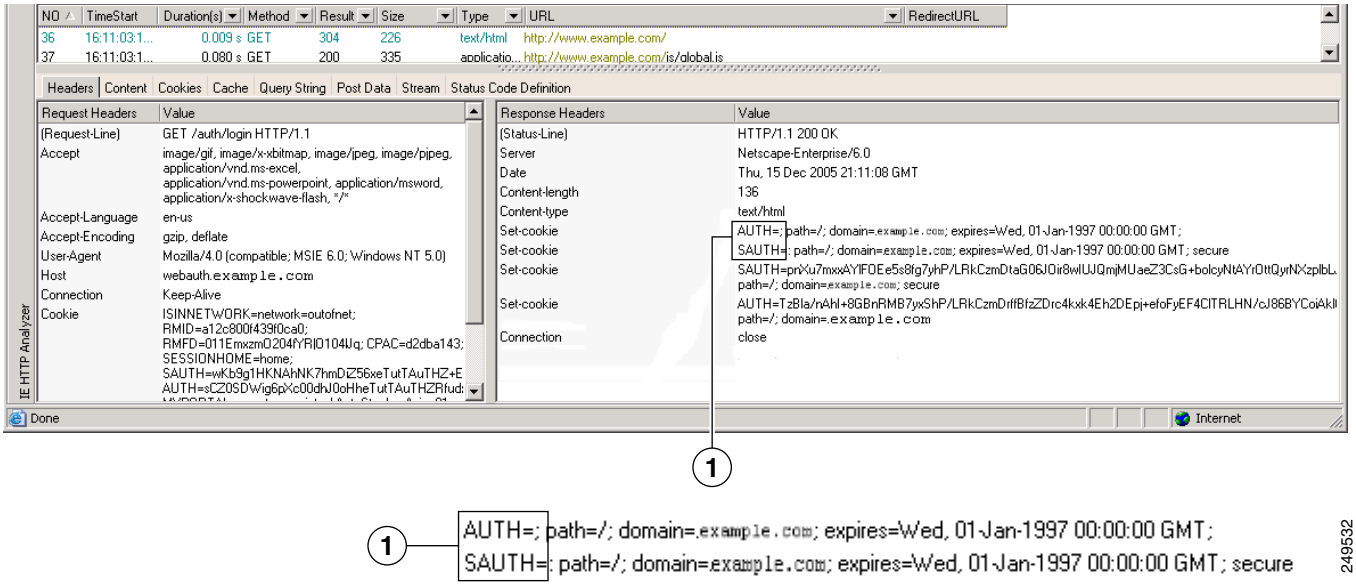
Step 6 If you successfully log in to the web server, examine the server response with the HTTP header analyzer to locate the name of the session cookie set by the server in your browser. This is the **auth-cookie-name** parameter.

In the following server response header, the name of the session cookie is SMSESSION. You just need the name, not the value.

```
Set-Cookie:
SMSESSION=yN4Yp5hHVNDgs4FT8dn7+Rwev41hsE49XlKc+1twie0gqnjbhkTkUnR8XWP3hvDH6PZP
bHIHtWLDKTa8ngDB/lbYTjIxrbDx8WPWwaG3CxVa3adOxHFR8yjD55GevK3ZF4ujgU1lhO6fta0dSS
OSepWvnsCb7IFxCw+MGiw0o88uHa2t4l+SillqfJvcpuXfiIAO06D/gtDF40Ow5YKHEl2KhDEvv+yQ
zxwfEz2cl7Ef5iMr8LgGcDK7qvMcvrgUqx68JQOK2+RSwtHQ15bCZmsDU5vQVCvSQWC8OMHNGwpS25
3XwRLvd/h6S/tM0k98QMv+i3N8oOdj1V7flBqecH7+kVrU01F6oFzr0zM1kMyLr5HhlVDh7B0k9wp0
dUFZiAzaf43jupD5f6CEkuLeudYW1xgNzsR8eqtPK6t1gFJyOn0s7QdNQ7q9knsPJsekRAH9hrLBhW
BLTU/3B1QS94wEGD2YTuiW36TiP14hYwOlCAYRj2/bY3+lYzVu7EmzMQ+UefYxh4cF2gYD8RZL2Rwm
P9JV5148I3XBFPNUw/3V5jf7nRuLr/CdfK3OO8+Pa3V6/nNhokErSgyxjzMd88DVzM41LxxaUDhbcm
koHT9ImzBvKzJX0J+o7FoUDFOxEdIqlAN4GNqk49cpi2sXDbIarALp6B13+tbB4MlHGH+0CPscZXqo
i/kon9YmGauHyRs+0m6wthdlAmCnvlJCDfDoXtn8DpabgiW6VDTrvl3SGPyQtUv7Wdahuq5SxbUzjY
2JxQnrUtwB977NCzYu2sOtN+dsEReWJ6ueyJBbMzKyzUB4L3i5uSYN50B4PCv1w5KdRKa5p3N0Nfq6
RM6dfipMEJw0Ny1sZ7ohz3fbvQ/YZ7lw/k7ods/8VbaR15ivkE8dSCzuf/AInHtCzuQ6wApzEp9CUo
G8/dapWriHjNoi4llJOgCst33wEhxFxcWy2UWxs4EZSjsI5GyBnefSQTPVfma5dc/emWor9vWr0HnT
QaHP5rg5dTNqunkDEdMIHfbeP3F90cZejVzihM6igiS6P/CEJAjE;Domain=.example.com;Path=
/
```

Figure 39-4 shows an example of authorization cookies in HTTP analyzer output. This is only an example; output varies widely across different websites.

Figure 39-4 Authorization cookies in sample HTTP analyzer output



1	Authorization cookies

Step 7 In some cases, the server may set the same cookie regardless of whether the authentication was successful or not, and such a cookie is unacceptable for SSO purposes. To confirm that the cookies are different, repeat Step 1 through Step 6 using invalid login credentials and then compare the “failure” cookie with the “success” cookie.

You now have the necessary parameter data to configure the security appliance for SSO with HTTP Form protocol.

Task Overview: Configuring SSO with HTTP Form Protocol

This section presents an overview of configuring SSO with the HTTP Form protocol.To enable SSO using HTTP Forms, perform the following tasks:

- Configure the uniform resource identifier on the authenticating web server to receive and process the form data (**action-uri**).
- Configure the username parameter (**user-parameter**).
- Configure the user password parameter (**password-parameter**).

You might also need to do the following tasks depending upon the requirements of authenticating web server:

- Configure a starting URL if the authenticating web server requires a pre-login cookie exchange (**start-url**).
- Configure any hidden authentication parameters required by the authenticating web server (**hidden-parameter**).
- Configure the name of an authentication cookie set by the authenticating web server (**auth-cookie-name**).

Detailed Tasks: Configuring SSO with HTTP Form Protocol

This section presents the detailed tasks required to configure SSO with the HTTP Form protocol. Perform the following steps to configure the security appliance to use HTTP Form protocol for SSO:

Step 1 If the authenticating web server requires it, enter the **start-url** command in aaa-server-host configuration mode to specify the URL from which to retrieve a pre-login cookie from the authenticating web server. For example, to specify the authenticating web server URL http://example.com/east/Area.do?Page-Grp1 in the testgrp1 server group with an IP address of 10.0.0.2, enter the following:

```
hostname(config)# aaa-server testgrp1 host 10.0.0.2
hostname(config-aaa-server-host)# start-url http://example.com/east/Area.do?Page-Grp1
hostname(config-aaa-server-host)#
```

Step 2 To specify a URI for an authentication program on the authenticating web server, enter the **action-uri** command in aaa-server- host configuration mode. A URI can be entered on multiple, sequential lines. The maximum number of characters per line is 255. The maximum number of characters for a complete URI is 2048. An example action URI follows:

```
http://www.example.com/auth/index.html/appdir/authc/forms/MCOlogin.fcc?TYPE=33554433&RE
ALMOID=06-000a1311-a828-1185-ab41-8333b16a0008&GUID=&SMAUTHREASON=0&METHOD=GET&SMAGENTN
AME=$SM$5FZmjnk3DRNwNjk2KcqVCFbIrNT9%2bJ0H0KPshFtg6rB1UV2PxkHqLw%3d%3d&TARGET=https%3A%
2F%2Fauth.example.com
```

To specify this action URI, enter the following commands:

```
hostname(config-aaa-server-host)# action-uri http://www.example.com/auth/index.htm
hostname(config-aaa-server-host)# action-uri l/appdir/authc/forms/MCOlogin.fcc?TYP
hostname(config-aaa-server-host)# action-uri 554433&REALMOID=06-000a1311-a828-1185
hostname(config-aaa-server-host)# action-uri -ab41-8333b16a0008&GUID=&SMAUTHREASON
hostname(config-aaa-server-host)# action-uri =0&METHOD=GET&SMAGENTNAME=$SM$5FZmjnk
hostname(config-aaa-server-host)# action-uri 3DRNwNjk2KcqVCFbIrNT9%2bJ0H0KPshFtg6r
hostname(config-aaa-server-host)# action-uri B1UV2PxkHqLw%3d%3d&TARGET=https%3A%2F
```

```
hostname(config-aaa-server-host)# action-uri %2Fauth.example.com
hostname(config-aaa-server-host)#
```

Note You must include the hostname and protocol in the action URI. In the preceding example, these appear at the start of the URI in http://www.example.com.

Step 3 To configure a username parameter for the HTTP POST request, enter the **user-parameter** command in aaa-server-host configuration mode. For example, the following command configures the username parameter userid:

```
hostname(config-aaa-server-host)# user-parameter userid
hostname(config-aaa-server-host)#
```

Step 4 To configure a user password parameter for the HTTP POST request, use the **password-parameter** command in aaa-server-host configuration mode. For example, the following command configures a user password parameter named user_password:

```
hostname(config-aaa-server-host)# password-parameter user_password
hostname(config-aaa-server-host)#
```

Step 5 To specify hidden parameters for exchange with the authenticating web server, use the **hidden-parameter** command in aaa-server-host configuration mode. An example hidden parameter excerpted from a POST request follows:

SMENC=ISO-8859-1&SMLOCALE=US-EN&target=https%3A%2F%2Fwww.example.com%2Femco%2Fappdir%2FAreaRoot.do%3FEMCOPageCode%3DENG&smauthreason=0

This hidden parameter includes four form entries and their values, separated by &. The four entries and their values are:

- SMENC with a value of ISO-8859-1
- SMLOCALE with a value of US-EN
- target with a value of https%3A%2F%2Fwww.example.com%2Femco%2Fappdir%2FAreaRoot.do
- %3FEMCOPageCode%3DENG
- smauthreason with a value of 0

To specify this hidden parameter, enter the following commands:

```
hostname(config)# aaa-server testgrp1 host example.com
hostname(config-aaa-server-host)# hidden-parameter SMENC=ISO-8859-1&SMLOCALE=US-EN&targe
hostname(config-aaa-server-host)# hidden-parameter t=https%3A%2F%2Fwww.example.com%2Femc
hostname(config-aaa-server-host)# hidden-parameter o%2Fappdir%2FAreaRoot.do%3FEMCOPageCo
hostname(config-aaa-server-host)# hidden-parameter de%3DENG&smauthreason=0
hostname(config-aaa-server-host)#
```

Step 6 To specify the name for the authentication cookie, enter the **auth-cookie-name** command in aaa-server-host configuration mode. This command is optional. The following example specifies the authentication cookie name of SsoAuthCookie:

```
hostname(config-aaa-server-host)# auth-cookie-name SsoAuthCookie
hostname(config-aaa-server-host)#
```

Authenticating with Digital Certificates

Clientless SSL VPN users that authenticate using digital certificates do not use global authentication and authorization settings. Instead, they use an authorization server to authenticate once the certificate validation occurs. For more information on authentication and authorization using digital certificates, see "Using Certificates and User Login Credentials" in the "Configuring AAA Servers and the Local Database" chapter.

Creating and Applying Clientless SSL VPN Resources

Creating and applying policies for clientless SSL VPN that govern access to resources at the central site includes the following task:

- Assigning Users to Group Policies

Chapter 32, "Configuring Connection Profiles, Group Policies, and Users" includes step-by-step instructions for all of these tasks.

Assigning Users to Group Policies

Assigning users to group policies simplifies the configuration by letting you apply policies to many users. You can use an internal authentication server or a RADIUS server to assign users to group policies. See Chapter 32, "Configuring Connection Profiles, Group Policies, and Users"for a thorough explanation of ways to simplify configuration with group policies.

Using the Security Appliance Authentication Server

You can configure users to authenticate to the security appliance internal authentication server, and assign these users to a group policy on the security appliance.

Using a RADIUS Server

Using a RADIUS server to authenticate users, assign users to group policies by following these steps:

Step 1 Authenticate the user with RADIUS and use the Class attribute to assign that user to a particular group policy.

Step 2 Set the class attribute to the group policy name in the format OU=group_name

For example, to assign a user of clientless SSL VPN to the SSL_VPN group, set the RADIUS Class Attribute to a value of *OU=SSL_VPN;* (Do not omit the semicolon.)

Configuring Connection Profile Attributes for Clientless SSL VPN

Table 39-1 provides a list of connection profile attributes that are specific to clientless SSL VPN. In addition to these attributes, you configure general connection profile attributes common to all VPN connections. For step-by-step information on configuring connection profiles, see "Configuring Connection Profiles for Clientless SSL VPN Sessions" in Chapter 32, "Configuring Connection Profiles, Group Policies, and Users."

Note In earlier releases, "connection profiles" were known as "tunnel groups." You configure a connection profile with tunnel-group commands. This chapter often uses these terms interchangeably.

Table 39-1 Connection Profile Attributes for Clientless SSL VPN

Command	Function
authentication	Sets the authentication method.
customization	Identifies the name of a previously defined customization to apply.
nbns-server	Identifies the name of the NetBIOS Name Service server (nbns-server) to use for CIFS name resolution.
group-alias	Specifies the alternate names by which the server can refer to a connection profile
group-url	Identifies one or more group URLs. If you configure this attribute, users coming in on a specified URL need not select a group at login
dns-group	Identifies the DNS server group that specifies the DNS server name, domain name, name server, number of retries, and timeout values
hic-fail-group-policy	Specifies a VPN feature policy if you use the Cisco Secure Desktop Manager to set the Group-Based Policy attribute to "Use Failure Group-Policy" or "Use Success Group-Policy, if criteria match."
override-svc-download	Overrides downloading the group-policy or username attributes configured for downloading the AnyConnect VPN client to the remote user.
radius-reject-message	Enables the display of the RADIUS reject message on the login screen when authentication is rejected.

Configuring Group Policy and User Attributes for Clientless SSL VPN

Table 39-2 provides a list of group policy and user attributes for clientless SSL VPN. For step-by-step instructions on configuring group policy and user attributes, see "Configuring Group Policies" and "Configuring Attributes for Specific Users" in Chapter 32, "Configuring Connection Profiles, Group Policies, and Users."

Table 39-2 Group Policy and User Attributes for Clientless SSL VPN

Command	Function
activex-relay	Lets a user who has established a clientless SSL VPN session use the browser to launch Microsoft Office applications. The applications use the session to download and upload Microsoft Office documents. The ActiveX relay remains in force until the clientless SSL VPN session closes.
auto-signon	Sets values for auto signon, which requires only that the user enter username and password credentials only once for a clientless SSL VPN connection.
customization	Assigns a customization object to a group-policy or user.
deny-message	Specifies the message delivered to a remote user who logs into clientless SSL VPN successfully, but has no VPN privileges.
file-browsing	Enables CIFS file browsing for file servers and shares. Browsing requires NBNS (Master Browser or WINS)
file-entry	Allows users to enter file server names to access.
filter	Sets the name of the webtype access list.
hidden-shares	Controls the visibility of hidden shares for CIFS files.
homepage	Sets the URL of the web page that displays upon login.
html-content-filter	Configures the content and objects to filter from the HTML for this group policy.
http-comp	Configures compression.
http-proxy	Configures the security appliance to use an external proxy server to handle HTTP requests.
keep-alive-ignore	Sets the maximum object size to ignore for updating the session timer.
port-forward	Applies a list of clientless SSL VPN TCP ports to forward. The user interface displays the applications on this list.
post-max-size	Sets the maximum object size to post.
smart-tunnel	Configures a list of programs to use smart tunnel.
sso-server	Sets the name of the SSO server.
storage-objects	Configures storage objects for the data stored between sessions.
svc	Configures SSL VPN Client attributes.
unix-auth-gid	Sets the UNIX group ID.
unix-auth-uid	Sets the UNIX user ID.
upload-max-size	Sets the maximum object size to upload.
url-entry	Controls the ability of the user to enter any HTTP/HTTP URL.
url-list	Applies a list of servers and URLs that Clientless SSL VPN portal page displays for end user access.
user-storage	Configures a location for storing user data between sessions.

Configuring Browser Access to Client-Server Plug-ins

The following sections describe the integration of browser plug-ins for clientless SSL VPN browser access:

- Introduction to Browser Plug-Ins, page 39-24
- Plug-in Requirements and Restrictions, page 39-25
- Preparing the Security Appliance for a Plug-in, page 39-25
- Installing Plug-ins Redistributed By Cisco, page 39-26
- Providing Access to Third-Party Plug-ins, page 39-28
- Viewing the Plug-ins Installed on the Security Appliance, page 39-29

Introduction to Browser Plug-Ins

A browser plug-in is a separate program that a web browser invokes to perform a dedicated function, such as connect a client to a server within the browser window. The security appliance lets you import plug-ins for download to remote browsers in clientless SSL VPN sessions. Of course, Cisco tests the plug-ins it redistributes, and in some cases, tests the connectivity of plug-ins we cannot redistribute. However, we do not recommend importing plug-ins that support streaming media at this time.

Note Per the GNU General Public License (GPL), Cisco redistributes plug-ins without having made any changes to them. Per the GPL, Cisco cannot directly enhance these plug-ins.

The security appliance does the following when you install a plug-in onto the flash device:

- (Cisco-distributed plug-ins only) Unpacks the jar file specified in the *URL*.
- Writes the file to the `csco-config/97/plugin` directory on the security appliance file system.
- Populates the drop-down menu next to the URL attributes in ASDM.
- Enables the plug-in for all future clientless SSL VPN sessions, and adds a main menu option and an option to the drop-down menu next to the Address field of the portal page.

  Table 39-3 shows the changes to the main menu and address field of the portal page when you add the plug-ins described in the following sections.

Table 39-3 Effects of Plug-ins on the Clientless SSL VPN Portal Page

Plug-in	Main Menu Option Added to Portal Page	Address Field Option Added to Portal Page
ica	Citrix Client	citrix://
rdp	Terminal Servers	rdp://
rdp2	Terminal Servers Vista	rdp2://
ssh,telnet	SSH	ssh://
	Telnet	telnet://
vnc	VNC Client	vnc://

When the user in a clientless SSL VPN session clicks the associated menu option on the portal page, the portal page displays a window to the interface and displays a help pane. The user can select the protocol displayed in the drop-down menu and enter the URL in the Address field to establish a connection.

Note Some Java plug-ins may report a status of connected or online even when a session to the destination service is not set up. The open-source plug-in reports the status, not the security appliance.

Plug-in Requirements and Restrictions

Clientless SSL VPN must be enabled on the security appliance to provide remote access to the plug-ins.

The plug-ins do not work if the security appliance configures the clientless session to use a proxy server.

Note The remote desktop protocol plug-in does not support load balancing with a session broker. Because of the way the protocol handles the redirect from the session broker, the connection fails. If a session broker is not used, the plug-in works.

The plug-ins support single sign-on (SSO). They use the *same* credentials entered to open the Clientless SSL VPN session. Because the plug-ins do not support macro substitution, you do not have the options to perform SSO on different fields such as the internal domain password or on an attribute on a Radius or LDAP server.

To configure SSO support for a plug-in, you install the plug-in, add a bookmark entry to display a link to the server, and specify SSO support when adding the bookmark.

The minimum access rights required for remote use belong to the guest privilege mode.

A stateful failover does not retain sessions established using plug-ins. Users must reconnect following a failover.

Preparing the Security Appliance for a Plug-in

Before installing a plug-in, prepare the security appliance as follows:

Step 1 Make sure clientless SSL VPN (“webvpn”) is enabled on a security appliance interface. To do so, enter the **show running-config** command.

Step 2 Install an SSL certificate onto the security appliance interface to which remote users use a fully-qualified domain name (FQDN) to connect.

Note Do not specify an IP address as the common name (CN) for the SSL certificate. The remote user attempts to use the FQDN to communicate with the security appliance. The remote PC must be able to use DNS or an entry in the System32\drivers\etc\hosts file to resolve the FQDN.

Go to the section that identifies the type of plug-in you want to provide for clientless SSL VPN access.

- Installing Plug-ins Redistributed By Cisco, page 39-26
- Providing Access to Third-Party Plug-ins, page 39-28

Installing Plug-ins Redistributed By Cisco

Cisco redistributes the following open-source, Java-based components to be accessed as plug-ins for web browsers in clientless SSL VPN sessions:

Table 39-4 Plug-ins Redistributed by Cisco

Cisco Download Link	Protocol	Description	Source of Redistributed Plug-in
rdp2-plugin.090211.jar	RDP2	Accesses Microsoft Terminal Services hosted by Windows Vista and Windows 2003 R2. Supports Remote Desktop ActiveX Control. **Note:** You can import the RDP and RDP2 plug-ins to make both of them available to clientless users.	Cisco redistributes this plug-in without any changes to it per the GNU General Public License. The original source of the redistributed plug-in is http://properjavardp.sourceforge.net/
rdp-plugin.080506.jar	RDP	Accesses Microsoft Terminal Services hosted by Windows 2003 R1. Supports Remote Desktop ActiveX Control.	Cisco redistributes this plug-in without any changes to it per the GNU General Public License. The source of the redistributed plug-in is http://properjavardp.sourceforge.net/
ssh-plugin.080430.jar	SSH	The Secure Shell-Telnet plug-in lets the remote user establish a Secure Shell or Telnet connection to a remote computer.	Cisco redistributes this plug-in without any changes to it per the GNU General Public License. The web site containing the source of the redistributed plug-in is http://javassh.org/
vnc-plugin.080130.jar	VNC	The Virtual Network Computing plug-in lets the remote user use a monitor, keyboard, and mouse to view and control a computer with remote desktop sharing turned on. This version changes the default color of the text, and contains updated French and Japanese help files.	Cisco redistributes this plug-in without any changes to it per the GNU General Public License. The web site containing the source of the redistributed plug-in is http://www.tightvnc.com/

These plug-ins are available on the Cisco Adaptive Security Appliance Software Download site.

Before installing a plug-in:

- Make sure clientless SSL VPN (“webvpn”) is enabled on an interface on the security appliance. To do so, enter the **show running-config** command.
- Create a temporary directory named “plugins” on a local TFTP or FTP server (for example, with the hostname “local_tftp_server”), and download the plug-ins from the Cisco web site to the “plugins” directory.

To provide clientless SSL VPN browser access to a plug-in redistributed by Cisco, install the plug-in onto the flash device of the security appliance by entering the following command in privileged EXEC mode.

import webvpn plug-in protocol *protocol URL*

protocol is one of the following values:

- **rdp** to provide plug-in access to Remote Desktop Protocol services. Then specify the path to the rdp-plugin.080130.jar file in the *URL* field.

- **ssh,telnet** to provide plug-in access to *both* Secure Shell and Telnet services. Then specify the path to the ssh-plugin.jar file in the *URL* field.

Caution Do *not* enter this command once for SSH and once for Telnet. When typing the **ssh,telnet** string, do *not* insert a space. Use the **revert webvpn plug-in protocol** command to remove any **import webvpn plug-in protocol** commands that deviate from these requirements.

- **vnc** to provide plug-in access to Virtual Network Computing services. Then specify the path to the vnc-plugin.080130.jar file in the *URL* field.

URL is the remote path to the source of the plug-in. Enter the host name or address of the TFTP or FTP server and the path to the plug-in.

The following example command adds clientless SSL VPN support for RDP:

```
hostname# import webvpn plug-in protocol rdp
tftp://local_tftp_server/plugins/rdp-plugin.080130.jar
Accessing
tftp://local_tftp_server/plugins/rdp-plugin.080130.jar...!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!
Writing file disk0:/csco_config/97/plugin/rdp...
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
329994 bytes copied in 5.350 secs (65998 bytes/sec)
```

The following example command adds clientless SSL VPN support for SSH and Telnet:

```
hostname# import webvpn plug-in protocol ssh,telnet
tftp://local_tftp_server/plugins/ssh-plugin.jar

Accessing
tftp://local_tftp_server/plugins/ssh-plugin.jar...!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Writing file disk0:/csco_config/97/plugin/ssh...
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
238510 bytes copied in 3.650 secs (79503 bytes/sec)
```

The following example command adds clientless SSL VPN support for VNC:

```
hostname# import webvpn plug-in protocol vnc
tftp://local_tftp_server/plugins/vnc-plugin.080130.jar

Accessing tftp://local_tftp_server/plugins/vnc-plugin.080130.jar...!!!!!!!!!!!!!!!!
Writing file disk0:/csco_config/97/plugin/vnc...
!!!!!!!!!!!!!!!!
58147 bytes copied in 2.40 secs (29073 bytes/sec)
```

Note The security appliance does not retain the **import webvpn plug-in protocol** command in the configuration. Instead, it loads the contents of the `csco-config/97/plugin` directory automatically. A secondary security appliance obtains the plug-ins from the primary security appliance.

After you import a plug-in, type the corresponding protocol and resource location in the address bar of the SSL VPN home page to access it. For example:

```
rdp://10.1.1.1
vnc://10.1.1.1
ssh://10.1.1.1
telnet://10.1.1.1
```

To disable and remove clientless SSL VPN support for a Java-based client application, as well as to remove it from the flash drive of the security appliance, use the following command:

revert webvpn plug-in protocol *protocol*

The following example command removes RDP:

```
hostname# revert webvpn plug-in protocol rdp
```

Providing Access to Third-Party Plug-ins

The open framework of the security appliance lets you add plug-ins to support third-party Java client/server applications.

Caution Cisco does not provide direct support for or recommend any particular plug-ins that are not redistributed by Cisco. As a provider of clientless SSL VPN services, you are responsible for reviewing and complying with any license agreements required for the use of plug-ins.

Example: Providing Access to a Citrix Java Presentation Server

The open framework of the security appliance lets you add plug-ins to support third-party Java client/server applications. As an example of how to provide Clientless SSL VPN browser access to third-party plug-ins, this section describes how to add Clientless SSL VPN support for the Citrix Presentation Server Client.

Caution Cisco does not provide direct support for or recommend any particular plug-ins that are not redistributed by Cisco. As a provider of Clientless SSL VPN services, you are responsible for reviewing and complying with any license agreements required for the use of plug-ins.

With a Citrix plug-in installed on the security appliance, Clientless SSL VPN users can use a connection to the security appliance to access Citrix MetaFrame services.

A stateful failover does not retain sessions established using the Citrix plug-in. Citrix users must reauthenticate after failover.

To provide access to the Citrix plug-in, follow the procedures in the following sections.

- Preparing the Citrix MetraFrame Server for Clientless SSL VPN Access
- Creating and Installing the Citrix Plug-in

Preparing the Citrix MetraFrame Server for Clientless SSL VPN Access

The security appliance performs the connectivity functions of the Citrix secure gateway when the Citrix client connects to the Citrix MetaFrame Server. Therefore, you must configure the Citrix Web Interface software to operate in a mode that does not use the (Citrix) “secure gateway.” Otherwise, the Citrix client cannot connect to the Citrix MetaFrame Server.

Note If you are not already providing support for a plug-in, you must follow the instructions in the“Preparing the Security Appliance for a Plug-in” section on page 39-25 before using this section.

Creating and Installing the Citrix Plug-in

To create and install the Citrix plug-in, perform the following steps:

Step 1 Download the ica-plugin.zip file from the Cisco Software Download web site.

This file contains files that Cisco customized for use with the Citrix plug-in.

Step 2 Download the Citrix Java client from the Citrix site.

Step 3 Extract the following files from the Citrix Java client, then add them to the ica-plugin.zip file:

- JICA-configN.jar
- JICAEngN.jar

You can use WinZip to perform this step.

Step 4 Ensure the EULA included with the Citrix Java client grants you the rights and permissions to deploy the client on your web servers.

Step 5 Open a CLI session with the security appliance and install the plug-in by entering the following command in privileged EXEC mode:

import webvpn plug-in protocol ica *URL*

URL is the host name or IP address and path to the ica-plugin.zip file.

Note After you import the plug-in, remote users can choose **ica** and enter *host***/?DesiredColor=4&DesiredHRes=1024&DesiredVRes=768** into the Address field of the portal page to access Citrix services. We recommend that you add a bookmark to make it easy for users to connect. Adding a bookmark is required if you want to provide SSO support for Citrix sessions.

Step 6 Establish an SSL VPN Clientless session and click the bookmark or enter the URL for the citrix server.

Use the Client for Java Administrator's Guide as needed.

Viewing the Plug-ins Installed on the Security Appliance

Enter the following command in privileged EXEC mode to list the Java-based client applications available to users of clientless SSL VPN:

show import webvpn plug-in

For example:

```
hostname# show import webvpn plug-in
ssh
rdp
vnc
ica
```

Configuring Application Access

The following sections describe how to enable smart tunnel access and port forwarding on clientless SSL VPN sessions, specify the applications to be provided with such access, and provide notes on using it:

- Configuring Smart Tunnel Access
- Configuring Port Forwarding
- Application Access User Notes

Configuring Smart Tunnel Access

The following sections describe smart tunnels and how to configure them:

- About Smart Tunnels
- Why Smart Tunnels?
- Smart Tunnel Requirements, Restrictions, and Limitations
- Adding Applications to Be Eligible for Smart Tunnel Access
- Assigning a Smart Tunnel List
- Configuring Smart Tunnel Auto Sign-on
- Automating Smart Tunnel Access
- Enabling and Disabling Smart Tunnel Access

About Smart Tunnels

A smart tunnel is a connection between a TCP-based application and a private site, using a clientless (browser-based) SSL VPN session with the security appliance as the pathway, and the security appliance as a proxy server. You can identify applications to which you want to grant smart tunnel access, and specify the local path to each application. For applications running on Microsoft Windows, you can also require a match of the SHA-1 hash of the checksum as a condition for granting smart tunnel access.

Lotus SameTime and Microsoft Outlook Express are examples of applications to which you might want to grant smart tunnel access.

Configuring smart tunnels requires one of the following procedures, depending on whether the application is a client or is a web-enabled application:

- Create one or more smart tunnel lists of the client applications, then assign the list to the group policies or local user policies for whom you want to provide smart tunnel access.
- Create one or more bookmark list entries that specify the URLs of the web-enabled applications eligible for smart tunnel access, then assign the list to the DAPs, group policies, or local user policies for whom you want to provide smart tunnel access.

You can also list web-enabled applications for which to automate the submission of login credentials in smart tunnel connections over clientless SSL VPN sessions.

Why Smart Tunnels?

Smart tunnel access lets a client TCP-based application use a browser-based VPN connection to connect to a service. It offers the following advantages to users, compared to plug-ins and the legacy technology, port forwarding:

- Smart tunnel offers better performance than plug-ins.
- Unlike port forwarding, smart tunnel simplifies the user experience by not requiring the user connection of the local application to the local port.
- Unlike port forwarding, smart tunnel does not require users to have administrator privileges.

The advantage of a plug-in is that it does not require the client application to be installed on the remote computer.

Smart Tunnel Requirements, Restrictions, and Limitations

The following sections categorize the smart tunnel requirements and limitations.

General Requirements and Limitations

Smart tunnel has the following general requirements and limitations:

- The remote host originating the smart tunnel must be running a 32-bit version of Microsoft Windows Vista, Windows XP, or Windows 2000; or Mac OS 10.4 or 10.5.
- Smart tunnel auto sign-on supports only Microsoft Internet Explorer on Windows.
- The browser must be enabled with Java, Microsoft ActiveX, or both.
- Smart tunnel supports only proxies placed between computers running Microsoft Windows and the security appliance. Smart tunnel uses the Internet Explorer configuration (that is, the one intended for system-wide use in Windows). If the remote computer requires a proxy server to reach the security appliance, the URL of the terminating end of the connection must be in the list of URLs excluded from proxy services. If the proxy configuration specifies that traffic destined for the ASA goes through a proxy, all smart tunnel traffic goes through the proxy.

  In an HTTP-based remote access scenario, sometimes a subnet does not provide user access to the VPN gateway. In this case, a proxy placed in front of the ASA to route traffic between the web and the end user's location provides web access. However, only VPN users can configure proxies placed in front of the ASA. When doing so, they must make sure these proxies support the CONNECT method. For proxies that require authentication, smart tunnel supports only the basic digest authentication type.
- When smart tunnel starts, the security appliance tunnels all traffic from the browser process the user used to initiate the clientless session. If the user starts another instance of the browser process, it passes all traffic to the tunnel. If the browser process is the same and the security appliance does not provide access to a given URL, the user cannot open it. As a workaround, the user can use a different browser from the one used to establish the clientless session.
- A stateful failover does not retain smart tunnel connections. Users must reconnect following a failover.

Windows Requirements and Limitations

The following requirements and limitations apply to Windows only:

- Only Winsock 2, TCP-based applications are eligible for smart tunnel access.

- The security appliance does not support the Microsoft Outlook Exchange (MAPI) proxy. Neither port forwarding nor the smart tunnel supports MAPI. For Microsoft Outlook Exchange communication using the MAPI protocol, remote users must use AnyConnect.
- Users of Microsoft Windows Vista who use smart tunnel or port forwarding must add the URL of the ASA to the Trusted Site zone. To access the Trusted Site zone, they must start Internet Explorer and choose the **Tools > Internet Options > Security** tab. Vista users can also disable Protected Mode to facilitate smart tunnel access; however, we recommend against this method because it increases vulnerability to attack.

Mac OS Requirements and Limitations

The following requirements and limitations apply to Mac OS only:

- Safari 3.1.1 or later, or Firefox 3.0 or later.
- Sun JRE 1.5 or later.
- Only applications started from the portal page can establish smart tunnel connections. This requirement includes smart tunnel support for Firefox. Using Firefox to start another instance of Firefox during the first use of a smart tunnel requires the user profile named csco_st. If this user profile is not present, the session prompts the user to create one.
- Applications using TCP that are dynamically linked to the SSL library can work over a smart tunnel.
- The PowerPC MAC operating system is not supported with smart tunnel.
- Smart tunnel does not support the following on Mac OS:
  - Proxy services.
  - Auto sign-on.
  - Applications that use two-level name spaces.
  - Console-based applications, such as Telnet, SSH, and cURL.
  - Applications using dlopen or dlsym to locate libsocket calls.
  - Statically linked applications to locate libsocket calls.

Adding Applications to Be Eligible for Smart Tunnel Access

The clientless SSL VPN configuration of each security appliance supports *smart tunnel lists*, each of which identifies one or more applications eligible for smart tunnel access. Because each group policy or username supports only one smart tunnel list, you must group each set of applications to be supported into a smart tunnel list.

To add an entry to a list of applications that can use a clientless SSL VPN session to connect to private sites, enter the following command in webvpn configuration mode:

smart-tunnel list *list application path* [**platform** *OS*] [*hash*]

To remove an application from a list, use the **no** form of the command, specifying both the list and the name of the application.

no smart-tunnel list *list application*

To remove an entire list of applications from the security appliance configuration, use the **no** form of the command, specifying only the list.

no smart-tunnel list *list*

- *list* is the name for a list of applications or programs. Use quotation marks around the name if it includes a space. The string can be up to 64 characters. The CLI creates the list if it is not present in the configuration. Otherwise, it adds the entry to the list.

  Note To view the smart tunnel list entries in the SSL VPN configuration, enter the **show running-config webvpn** command in privileged EXEC mode.

- *application* is a string that serves as a unique index to each entry in the smart tunnel list. It typically names the application to be granted smart tunnel access. To support multiple versions of an application for which you choose to specify different paths or hash values, you can use this attribute to differentiate entries, specifying the OS, and name and version of the application supported by each list entry. The string can be up to 64 characters. To change an entry already present in a smart tunnel list, enter the name of the entry to be changed.
- *path* is the filename and extension of the application; or a path to the application, including its filename and extension. The string can be up to 128 characters.

  Windows requires an exact match of this value to the right side of the application path on the remote host to qualify the application for smart tunnel access. If you specify only the filename for Windows, SSL VPN does not enforce a location restriction on the remote host to qualify the application for smart tunnel access.

  If you specify a path and the user installed the application in another location, that application does not qualify. The application can reside on any path as long as the right side of the string matches the value you enter.

  To authorize an application for smart tunnel access if it is present on one of several paths on the remote host, either specify only the name and extension of the application when you enter the *path* value; or enter the **smart-tunnel list** command once for each path, entering the same *list* string, but specifying the unique *application* string and *path* value in each command.

  Note A sudden problem with smart tunnel access may be an indication that a *Process Name* value is not up-to-date with an application upgrade. For example, the default path to an application sometimes changes following the acquisition of the company that produces the application and the next application upgrade.

  For Windows, if you want to add smart tunnel access to an application started from the command prompt, you must specify “cmd.exe” in the Process Name of one entry in the smart tunnel list, and specify the path to the application itself in another entry, because “cmd.exe” is the parent of the application.

  Mac OS requires the full path to the process, and is case-sensitive. To avoid specifying a path for each username, insert a tilde (~) before the partial path (e.g., ~/bin/vnc).
- **platform** is **windows** or **mac** to indicate the host OS of the application. The default value is **platform windows**.
- *hash* (Optional) To obtain this value, enter the checksum of the application (that is, the checksum of the executable file) into a utility that calculates a hash using the SHA-1 algorithm. One example of such a utility is the Microsoft File Checksum Integrity Verifier (FCIV), which is available at http://support.microsoft.com/kb/841290/. After installing FCIV, place a temporary copy of the application to be hashed on a path that contains no spaces (for example, c:/fciv.exe), then enter **fciv.exe -sha1** *application* at the command line (for example, **fciv.exe -sha1 c:\msimn.exe**) to display the SHA-1 hash.

The SHA-1 hash is always 40 hexadecimal characters.

Before authorizing an application for smart tunnel access, clientless SSL VPN calculates the hash of the application matching the *path*. It qualifies the application for smart tunnel access if the result matches the value of *hash*.

Entering a hash provides a reasonable assurance that SSL VPN does not qualify an illegitimate file that matches the string you specified in the *path*. Because the checksum varies with each version or patch of an application, the *hash* you enter can only match one version or patch on the remote host. To specify a *hash* for more than one version of an application, enter the **smart-tunnel list** command once for each version, entering the same *list* string, but specifying a unique *application* string and a unique *hash* value.

Note You must maintain the smart tunnel list in the future if you enter *hash* values and you want to support future versions or patches of an application with smart tunnel access. A sudden problem with smart tunnel access may be an indication that the application list containing *hash* values is not up-to-date with an application upgrade. You can avoid this problem by not entering a *hash*.

Table 39-5 Example smart-tunnel Commands

Function	OS	Commands
Add Lotus SameTime to a smart tunnel list named lotus.	Windows (default platform)	`smart-tunnel list lotus LotusSametime connect.exe`
Add the Lotus 6.0 thick client with Domino Server 6.5.5	Windows	`smart-tunnel list lotus lotusnotes notes.exefs` `smart-tunnel list lotus lotusnlnotes nlnotes.exe` `smart-tunnel list lotus lotusntaskldr ntaskldr.exe` `smart-tunnel list lotus lotusnfileret nfileret.exe`
Add the command prompt to a smart tunnel list named apps. **Note**: This is necessary to provide smart tunnel access to a Microsoft Windows application started from the command prompt. You must also add the application itself to the list.	Windows	`smart-tunnel list apps CommandPrompt cmd.exe`
Add Windows Outlook Express.	Windows	`smart-tunnel list apps OutlookExpress msimn.exe`
Add Windows Outlook Express, permitting smart tunnel support for it only if its path on the remote host matches the string.	Windows	`smart-tunnel list apps OutlookExpress "\Program Files\Outlook Express\msimn.exe"`
Add Windows Outlook Express, permitting smart tunnel support for it only if its hash matches the string.	Windows	`smart-tunnel list apps OutlookExpress msimn.exe 4739647b255d3ea865554e27c3f96b9476e75061`
Add Safari, permitting smart tunnel support for it only if its path on the remote host matches the string.	Mac OS	`smart-tunnel list apps Safari "/Applications/Safari" platform mac`
Add smart tunnel support for a new Terminal window.	Mac OS	`smart-tunnel list apps Terminal terminal platform mac`

Table 39-5 Example smart-tunnel Commands

Function	OS	Commands
Add smart tunnel support for an application started from a Mac Terminal window. All words after Terminal inside the quotation marks enter the command line.	Mac OS	`smart-tunnel list apps Terminal "terminal open -a MacTelnet" platform mac`
Add smart tunnel support for VNC, regardless of the user path to the VNC executable file.	Mac OS	`smart-tunnel list apps vnc "~/bin/vnc" platform mac`

Following the configuration of a smart tunnel list, assign the list to group policies or usernames, as described in the next section.

Assigning a Smart Tunnel List

For each group policy and username, you can configure clientless SSL VPN to do one of the following:

- Start smart tunnel access automatically upon user login.
- Enable smart tunnel access upon user login, but require the user to start it manually, using the **Application Access > Start Smart Tunnels** button on the clientless SSL VPN Portal Page.

Note These options are mutually exclusive for each group policy and username. Use only one.

Table 39-6 lists the smart tunnel commands available to each group policy and username. The configuration of each group policy and username supports only one of these commands at a time, so when you enter one, the security appliance replaces the one present in the configuration of the group policy or username in question with the new one, or in the case of the last command, simply removes the smart-tunnel command already present in the group policy or username.

Table 39-6 group-policy and username webvpn Smart Tunnel Commands

Command	Description
smart-tunnel auto-start *list*	Starts smart tunnel access automatically upon user login.
smart-tunnel enable *list*	Enables smart tunnel access upon user login, but requires the user to start smart tunnel access manually, using the **Application Access > Start Smart Tunnels** button on the clientless SSL VPN portal page.
smart-tunnel disable	Prevents smart tunnel access.
no smart-tunnel [**auto-start** *list* | **enable** *list* | **disable**]	Removes a **smart-tunnel** command from the group policy or username configuration, which then inherits the [**no**] **smart-tunnel** command from the default group-policy. The keywords following the **no smart-tunnel** command are optional, however, they restrict the removal to the named smart-tunnel command.

For details, go to the section that addresses the option you want to use.

Configuring Smart Tunnel Auto Sign-on

The following sections describe how to list the servers for which to provide auto sign-on in smart tunnel connections, and assign the lists to group policies or usernames.

Specifying Servers for Smart Tunnel Auto Sign-on

The Add Smart Tunnel Auto Sign-on Server List dialog box lets you add one or more lists of servers for which to automate the submission of login credentials during smart tunnel setup. The Edit Smart Tunnel Auto-signon Server List dialog box lets you modify the contents of these lists.

To create a list of servers for which to automate the submission of credentials in smart tunnel connections, enter the command in webvpn configuration mode.

[**no**] **smart-tunnel auto-signon** *list* [**use-domain**] {**ip** *ip-address* [*netmask*] | **host** *hostname-mask*}

Use this command for each server you want to add to a list. To remove an entry from a list, use the **no** form of the command, specifying both the list and the IP address or hostname, as it appears in the security appliance configuration. To display the smart tunnel auto sign-on list entries, enter the **show running-config webvpn smart-tunnel** command in privileged EXEC mode.

To remove an entire list of servers from the security appliance configuration, use the **no** form of the command, specifying only the list, as follows:

no smart-tunnel auto-signon *list*

- *list* names the list of remote servers. Use quotation marks around the name if it includes a space. The string can be up to 64 characters. The security appliance creates the list if it is not already present in the configuration. Otherwise, it adds the entry to the list. Assign a name that will help you to distinguish its contents or purpose from other lists are likely to be configured.
- **use-domain** (optional) adds the Windows domain to the username if authentication requires it. If you enter this keyword, be sure to specify the domain name when assigning the smart tunnel list to one or more group policies, or usernames.
- **ip** specifies the server by its IP address and netmask.
- *ip-address* [*netmask*] identifies the sub-network of hosts to auto-authenticate to.
- **host** specifies the server by its host name or wildcard mask. Using this option protects the configuration from dynamic changes to IP addresses.
- *hostname-mask* is the host name or wildcard mask to auto-authenticate to.

The following command adds all hosts in the subnet and adds the Windows domain to the username if authentication requires it:

```
asa2(config-webvpn)# smart-tunnel auto-signon HR use-domain ip 192.32.22.56 255.255.255.0
```

The following command removes that entry from the list:

```
asa2(config-webvpn)# no smart-tunnel auto-signon HR use-domain ip 192.32.22.56
255.255.255.0
```

The command shown above also removes the list named HR if the entry removed is the only entry in the list. Otherwise, the following command removes the entire list from the security appliance configuration:

```
asa2(config-webvpn)# no smart-tunnel auto-signon HR
```

The following command adds all hosts in the domain to the smart tunnel auto sign-on list named intranet:

```
asa2(config-webvpn)# smart-tunnel auto-signon intranet host *.exampledomain.com
```

The following command removes that entry from the list:

```
asa2(config-webvpn)# no smart-tunnel auto-signon intranet host *.exampledomain.com
```

Following the configuration of the smart tunnel auto sign-on server list, you must assign it to a group policy or a local user policy for it to become active, as described in the next section.

Adding or Editing a Smart Tunnel Auto Sign-on Server Entry

To enable smart tunnel auto sign-on in clientless (browser-based) SSL VPN sessions, use the **smart-tunnel auto-signon enable** command in group-policy webvpn configuration mode or username webvpn configuration mode.

[**no**] **smart-tunnel auto-signon enable** *list* [**domain** *domain*]

To remove the **smart-tunnel auto-signon enable** command from the group policy or username and inherit it from the default group-policy, use the **no** form of the command.

- *list* is the name of a smart tunnel auto sign-on list already present in the security appliance webvpn configuration. To view the smart tunnel auto sign-on list entries in the SSL VPN configuration, enter the **show running-config webvpn smart-tunnel** command in privileged EXEC mode.
- **domain** *domain* (optional) is the name of the domain to be added to the username during authentication. If you enter a domain, enter the **use-domain** keyword in the list entries.

The smart-tunnel auto sign-on feature supports only applications communicating HTTP and HTTPS using the Microsoft WININET library. For example, Microsoft Internet Explorer uses the WININET dynamic linked library to communicate with web servers.

You must use the **smart-tunnel auto-signon** *list* command to create a list of servers first. You can assign only one list to a group policy or username.

The following commands enable the smart tunnel auto sign-on list named HR:

```
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# smart-tunnel auto-signon enable HR
hostname(config-group-webvpn)
```

The following command enables the smart tunnel auto sign-on list named HR and adds the domain named CISCO to the username during authentication:

```
hostname(config-group-webvpn)# smart-tunnel auto-signon enable HR domain CISCO
```

The following command removes the smart tunnel auto sign-on list named HR from the group policy and inherits the smart tunnel auto sign-on list command from the default group policy:

```
hostname(config-group-webvpn)# no smart-tunnel auto-signon enable HR
```

Automating Smart Tunnel Access

To start smart tunnel access automatically upon user login, enter the following command in group-policy webvpn configuration mode or username webvpn configuration mode:

smart-tunnel auto-start *list*

list is the name of the smart tunnel list already present in the security appliance webvpn configuration. You cannot assign more than smart tunnel list to a group policy or username. To view the smart tunnel list entries in the SSL VPN configuration, enter the **show running-config webvpn** command in privileged EXEC mode.

To remove the **smart-tunnel** command from the group policy or username and inherit the [**no**] **smart-tunnel** command from the default group-policy, use the **no** form of the command.

no smart-tunnel

The following commands assign the smart tunnel list named apps1 to the group policy:

```
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# smart-tunnel auto-start apps1
```

Enabling and Disabling Smart Tunnel Access

By default, smart tunnels are disabled. If you enable smart tunnel access, the user will have to start it manually, using the **Application Access > Start Smart Tunnels** button on the clientless SSL VPN portal page. If you enter the **smart-tunnel auto-start** *list* command described in the previous section instead of the **smart-tunnel enable** *list* command, the user will not have to start smart tunnel access manually.

To enable smart tunnel access, enter the following command in group-policy webvpn configuration mode or username webvpn configuration mode:

smart-tunnel [**enable** *list* | **disable**]

list is the name of the smart tunnel list already present in the security appliance webvpn configuration. You cannot assign more than smart tunnel list to a group policy or username. To view the smart tunnel list entries in the SSL VPN configuration, enter the **show running-config webvpn** command in privileged EXEC mode.

To remove the **smart-tunnel** command from the group policy or local user policy, and inherit the [**no**] **smart-tunnel** command from the default group-policy, use the **no** form of the command.

no smart-tunnel

The following commands assign the smart tunnel list named apps1 to the group policy:

```
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# smart-tunnel enable apps1
```

The following command disables smart tunnel access:

```
hostname(config-group-webvpn)# smart-tunnel disable
```

Configuring Port Forwarding

The following sections describe port forwarding and how to configure it:

- About Port Forwarding
- Why Port Forwarding?
- Port Forwarding Requirements and Restrictions
- Configuring DNS for Port Forwarding
- Adding Applications to Be Eligible for Port Forwarding
- Assigning a Port Forwarding List
- Automating Port Forwarding
- Enabling and Disabling Port Forwarding

About Port Forwarding

Port forwarding lets users access TCP-based applications over a clientless SSL VPN connection. Such applications include the following:

- Lotus Notes
- Microsoft Outlook
- Microsoft Outlook Express
- Perforce
- Sametime
- Secure FTP (FTP over SSH)
- SSH
- TELNET
- Windows Terminal Service
- XDDTS

Other TCP-based applications may also work, but we have not tested them. Protocols that use UDP do not work.

Why Port Forwarding?

Port forwarding is the legacy technology for supporting TCP-based applications over a clientless SSL VPN connection. You may choose to use port forwarding because you have built earlier configurations that support this technology.

Please consider the following alternatives to port forwarding:

- Smart tunnel access offers the following advantages to users:
  - Smart tunnel offers better performance than plug-ins.
  - Unlike port forwarding, smart tunnel simplifies the user experience by not requiring the user connection of the local application to the local port.
  - Unlike port forwarding, smart tunnel does not require users to have administrator privileges.
- Unlike port forwarding and smart tunnel access, a plug-in does not require the client application to be installed on the remote computer.

When configuring port forwarding on the security appliance, you specify the port the application uses. When configuring smart tunnel access, you specify the name of the executable file or its path.

Port Forwarding Requirements and Restrictions

The following restrictions apply to port forwarding:

- The remote host must be running a 32-bit version of one of the following:
  - Microsoft Windows Vista, Windows XP SP2 or SP3; or Windows 2000 SP4.
  - Apple Mac OS X 10.4 or 10.5 with Safari 2.0.4(419.3).
  - Fedora Core 4
- The remote host must also be running Sun JRE 1.5 or later.

- Browser-based users of Safari on Mac OS X 10.5.3 must identify a client certificate for use with the URL of the security appliance, once with the trailing slash and once without it, because of the way Safari interprets URLs. For example,
  - https://example.com/
  - https://example.com

  For details, go to the Safari, Mac OS X 10.5.3: Changes in client certificate authentication.
- Users of Microsoft Windows Vista who use port forwarding or smart tunnels must add the URL of the ASA to the Trusted Site zone. To access the Trusted Site zone, they must start Internet Explorer and choose the **Tools > Internet Options > Security** tab. Vista users can also disable Protected Mode to facilitate smart tunnel access; however, we recommend against this method because it increases the computer's vulnerability to attack.
- Port forwarding supports only TCP applications that use static TCP ports. Applications that use dynamic ports or multiple TCP ports are not supported. For example, SecureFTP, which uses port 22, works over clientless SSL VPN port forwarding, but standard FTP, which uses ports 20 and 21, does not.
- Port forwarding does not support protocols that use UDP.
- The security appliance does not support the Microsoft Outlook Exchange (MAPI) proxy. For Microsoft Outlook Exchange communication using the MAPI protocol, remote users must use AnyConnect.
- A stateful failover does not retain sessions established using Application Access (either port forwarding or smart tunnel access). Users must reconnect following a failover.
- Port forwarding does not support connections to personal digital assistants.
- Because port forwarding requires downloading the Java applet and configuring the local client, and because doing so requires administrator permissions on the local system, it is unlikely that users will be able to use applications when they connect from public remote systems.

Caution Make sure Sun Microsystems Java Runtime Environment (JRE) 1.5.x or later is installed on the remote computers to support port forwarding (application access) and digital certificates. If JRE 1.4.x is running and the user authenticates with a digital certificate, the application fails to start because JRE cannot access the web browser certificate store.

  The Java applet displays in its own window on the end user HTML interface. It shows the contents of the list of forwarded ports available to the user, as well as which ports are active, and amount of traffic in bytes sent and received.
- Neither port forwarding nor the ASDM Java applet work with user authentication using digital certificates. Java does not have the ability to access the web browser keystore. Therefore Java cannot use certificates that the browser uses to authenticate users, and the application cannot start.

Configuring DNS for Port Forwarding

Port Forwarding forwards the domain name of the remote server or its IP address to the ASA for resolution and connection. In other words, the port forwarding applet accepts a request from the application and forwards it to the ASA. The ASA makes the appropriate DNS queries and establishes the connection on behalf of the port forwarding applet. The port forwarding applet only makes DNS queries to the ASA. It updates the host file so that when a port forwarding application attempts a DNS query, the query redirects to a loopback address. Configure the security appliance to accept the DNS requests from the port forwarding applet as follows:

Step 1 Use the **dns server-group** command in global configuration mode to enter the dns server-group mode; then use the **domain-name** command to specify the domain name and **name-server** command to resolve the domain name to an IP address. The default setting of domain-name is DefaultDNS.

The following example configures a DNS server group named example.com:

```
hostname(config)# dns server-group example.com
hostname(config-dns-server-group)# domain-name example.com
hostname(config-dns-server-group)# name-server 192.168.10.10
```

Step 2 (Required only if you are using a domain name other than the default one [DefaultDNS])—Use the **dns-group** command in tunnel-group webvpn configuration mode to specify the domain name the tunnel groups will use. By default, the security appliance assigns the DefaultWEBVPNGroup as the default tunnel group for clientless connections; follow this instruction if the security appliance uses that tunnel group to assign settings to the clientless connections. Otherwise, follow this step for each tunnel configured for clientless connections.

For example,

```
asa2(config-dns-server-group)# exit
asa2(config)# tunnel-group DefaultWEBVPNGroup webvpn-attributes
asa2(config-tunnel-webvpn)# dns-group example.com
```

Adding Applications to Be Eligible for Port Forwarding

The clientless SSL VPN configuration of each security appliance supports *port forwarding lists*, each of which specifies local and remote ports used by the applications for which you want to provide access. Because each group policy or username supports only one port forwarding list, you must group each set of applications to be supported into a list. To display the port forwarding list entries already present in the security appliance configuration, enter the following command in privileged EXEC mode:

show run webvpn port-forward

To add a port forwarding entry to a list, enter the following command in webvpn configuration mode:

port-forward {*list_name local_port remote_server remote_port description*}

list_name—Name for a set of applications (technically, a set of forwarded TCP ports) for users of clientless SSL VPN sessions to access. The security appliance creates a list using the name you enter if it does not recognize it. Otherwise, it adds the port forwarding entry to the list. Maximum 64 characters.

local_port—Port that listens for TCP traffic for an application running on the user's computer. You can use a local port number only once for each port forwarding list. Enter a port number in the range 1-65535 or port name. To avoid conflicts with existing services, use a port number greater than 1024.

remote_server—DNS name or IP address of the remote server for an application. The IP address can be in IPv4 or IPv6 format. We recommend a DNS name so that you do not have to configure the client applications for a specific IP address.

Caution The DNS name must match the one assigned to the tunnel group to establish the tunnel and resolve to an IP address, per the instructions in the previous section. The default setting for both the **domain-name group** and **dns-group** commands described in that section is DefaultDNS.

remote_port—Port to connect to for this application on the remote server. This is the actual port the application uses. Enter a port number in the range 1-65535 or port name.

description—Application name or short description that displays on the end user Port Forwarding Java applet screen. Maximum 64 characters.

To remove an entry from a list, use the **no** form of the command, specifying both the list and the local port. In this case, the remoteserver, remoteport, and description are optional.

no port-forward *list_name local_port*

The following table shows the values used for example applications.

Application	Local Port	Server DNS Name	Remote Port	Description
IMAP4S e-mail	20143	IMAP4Sserver	143	Get Mail
SMTPS e-mail	20025	SMTPSserver	25	Send Mail
DDTS over SSH	20022	DDTSserver	22	DDTS over SSH
Telnet	20023	Telnetserver	23	Telnet

The following example shows how to create a port forwarding list called *SalesGroupPorts* that provides access to these applications:

```
hostname(config)# webvpn
hostname(config-webvpn)# port-forward SalesGroupPorts 20143 IMAP4Sserver 143 Get Mail
hostname(config-webvpn)# port-forward SalesGroupPorts 20025 SMTPSserver 25 Send Mail
hostname(config-webvpn)# port-forward SalesGroupPorts 20022 DDTSserver 22 DDTS over SSH
hostname(config-webvpn)# port-forward SalesGroupPorts 20023 Telnetserver 23 Telnet
```

Following the configuration of a port forwarding list and assign the list to group policies or usernames, as described in the next section.

Assigning a Port Forwarding List

For each group policy and username, you can configure clientless SSL VPN to do one of the following:

- Start port forwarding access automatically upon user login.
- Enable port forwarding access upon user login, but require the user to start it manually, using the **Application Access > Start Applications** button on the clientless SSL VPN Portal Page.

Note These options are mutually exclusive for each group policy and username. Use only one.

Table 39-7 lists the **port-forward** commands available to each group policy and username. The configuration of each group policy and username supports only one of these commands at a time, so when you enter one, the security appliance replaces the one present in the configuration of the group policy or username in question with the new one, or in the case of the last command, simply removes the **port-forward** command from the group policy or username configuration.

Table 39-7 group-policy and username webvpn port-forward Commands

Command	Description
port-forward auto-start *list_name*	Starts port forwarding automatically upon user login.
port-forward enable *list_name*	Enables port forwarding upon user login, but requires the user to start port forwarding manually, using the **Application Access > Start Applications** button on the clientless SSL VPN portal page.
port-forward disable	Prevents port forwarding.
no port-forward [**auto-start** *list_name* | **enable** *list_name* | **disable**]	Removes a **port-forward** command from the group policy or username configuration, which then inherits the [**no**] **port-forward** command from the default group-policy. The keywords following the **no port-forward** command are optional, however, they restrict the removal to the named **port-forward** command.

For details, go to the section that addresses the option you want to use.

Automating Port Forwarding

To start port forwarding automatically upon user login, enter the following command in group-policy webvpn configuration mode or username webvpn configuration mode:

port-forward auto-start *list_name*

list_name names the port forwarding list already present in the security appliance webvpn configuration. You cannot assign more than one port forwarding list to a group policy or username. To display the port forwarding list entries present in the security appliance configuration, enter the **show run webvpn port-forward** command in privileged EXEC mode.

To remove the **port-forward** command from the group policy or username and inherit the [**no**] **port-forward** command from the default group-policy, use the **no** form of the command.

no port-forward

The following commands assign the port forwarding list named apps1 to the group policy:

```
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# port-forward auto-start apps1
```

Enabling and Disabling Port Forwarding

By default, port forwarding is disabled. If you enable port forwarding, the user will have to start it manually, using the **Application Access > Start Applications** button on the clientless SSL VPN portal page. If you enter the **port-forward auto-start** *list_name* command described in the previous section instead of the **port-forward enable** *list_name* command, the user will not have to start port forwarding manually to use it.

To enable or disable port forwarding, enter the following command in group-policy webvpn configuration mode or username webvpn configuration mode:

port-forward [**enable** *list_name* | **disable**]

list_name is the name of the port forwarding list already present in the security appliance webvpn configuration. You cannot assign more than one port forwarding list to a group policy or username. To view the port forwarding list entries, enter the **show running-config port-forward** command in privileged EXEC mode.

To remove the **port-forward** command from the group policy or username and inherit the [**no**] **port-forward** command from the default group-policy, use the **no** form of the command.

no port-forward

The following commands assign the port forwarding list named apps1 to the group policy:

```
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# port-forward enable apps1
```

The following command disables port forwarding:

```
hostname(config-group-webvpn)# port-forward disable
```

Application Access User Notes

The following sections provide information about using application access:

- Using Application Access on Vista
- Closing Application Access to Prevent hosts File Errors
- Recovering from hosts File Errors When Using Application Access

Note The security appliance does not support the Microsoft Outlook Exchange (MAPI) proxy. Neither the smart tunnel feature nor port forwarding supports MAPI. For Microsoft Outlook Exchange communication using the MAPI protocol, remote users must use AnyConnect.

Using Application Access on Vista

Users of Microsoft Windows Vista who use smart tunnels or port forwarding must add the URL of the ASA to the Trusted Site zone. To access the Trusted Site zone, they must start Internet Explorer and choose the Tools > Internet Options > Security tab. Vista users can also disable Protected Mode to facilitate smart tunnel access; however, we recommend against this method because it increases the computer's vulnerability to attack.

Closing Application Access to Prevent hosts File Errors

To prevent hosts file errors that can interfere with Application Access, close the Application Access window properly when you finish using Application Access. To do so, click the close icon.

Recovering from hosts File Errors When Using Application Access

The following errors can occur if you do not close the Application Access window properly:

- The next time you try to start Application Access, it might be disabled; you receive a `Backup HOSTS File Found` error message.

- The applications themselves might be disabled or might malfunction, even when you are running them locally.

These errors can result from terminating the Application Access window in any improper way. For example:

- Your browser crashes while you are using Application Access.
- A power outage or system shutdown occurs while you are using Application Access.
- You minimize the Application Access window while you are working, then shut down your computer with the window active (but minimized).

This section includes the following topics:

- Understanding the hosts File
- Stopping Application Access Improperly
- Reconfiguring a hosts File Automatically Using Clientless SSL VPN
- Reconfiguring hosts File Manually

Understanding the hosts File

The hosts file on your local system maps IP addresses to host names. When you start Application Access, clientless SSL VPN modifies the hosts file, adding clientless SSL VPN-specific entries. Stopping Application Access by properly closing the Application Access window returns the file to its original state.

Before invoking Application Access...	hosts file is in original state.
When Application Access starts....	• Clientless SSL VPN copies the hosts file to hosts.webvpn, thus creating a backup. • Clientless SSL VPN then edits the hosts file, inserting clientless SSL VPN-specific information.
When Application Access stops...	• Clientless SSL VPN copies the backup file to the `hosts` file, thus restoring the hosts file to its original state. • Clientless SSL VPN deletes hosts.webvpn.
After finishing Application Access...	hosts file is in original state.

Note Microsoft anti-spyware software blocks changes that the port forwarding Java applet makes to the hosts file. See www.microsoft.com for information on how to allow hosts file changes when using anti-spyware software.

Stopping Application Access Improperly

When Application Access terminates abnormally, the `hosts` file remains in a clientless SSL VPN-customized state. Clientless SSL VPN checks the state the next time you start Application Access by searching for a hosts.webvpn file. If it finds one, a `Backup HOSTS File Found` error message (Figure 39-5) appears, and Application Access is temporarily disabled.

Once you shut down Application Access improperly, you leave your remote access client/server applications in limbo. If you try to start these applications without using clientless SSL VPN, they might malfunction. You might find that hosts that you normally connect to are unavailable. This situation could commonly occur if you run applications remotely from home, fail to quit the Application Access window before shutting down the computer, then try to run the applications later from the office.

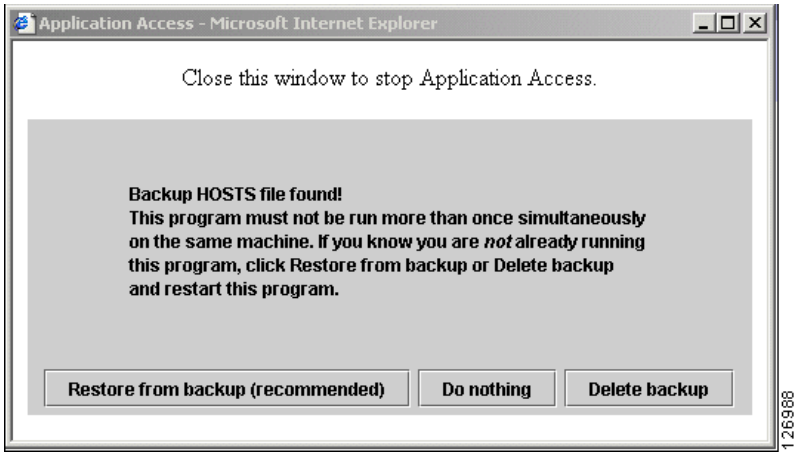
Reconfiguring a hosts File Automatically Using Clientless SSL VPN

If you are able to connect to your remote access server, follow these steps to reconfigure the hosts file and reenable both Application Access and the applications.

Step 1 Start clientless SSL VPN and log in. The home page opens.

Step 2 Click the **Applications Access** link. A `Backup HOSTS File Found` message appears. (See Figure 39-5.)

Figure 39-5 Backup HOSTS File Found Message



Step 3 Choose one of the following options:

- **Restore from backup**—Clientless SSL VPN forces a proper shutdown. It copies the hosts.webvpn backup file to the `hosts` file, restoring it to its original state, then deletes hosts.webvpn. You then have to restart Application Access.
- **Do nothing**—Application Access does not start. The remote access home page reappears.
- **Delete backup**—Clientless SSL VPN deletes the hosts.webvpn file, leaving the hosts file in its clientless SSL VPN-customized state. The original `hosts` file settings are lost. Application Access then starts, using the clientless SSL VPN-customized hosts file as the new original. Choose this option only if you are unconcerned about losing hosts file settings. If you or a program you use might have edited the hosts file after Application Access has shut down improperly, choose one of the other options, or edit the hosts file manually. (See "Reconfiguring hosts File Manually.")

Reconfiguring hosts File Manually

If you are not able to connect to your remote access server from your current location, or if you have customized the hosts file and do not want to lose your edits, follow these steps to reconfigure the hosts file and reenable both Application Access and the applications.

Step 1 Locate and edit your hosts file. The most common location is c:\windows\sysem32\drivers\etc\hosts.

Step 2 Check to see if any lines contain the string: `# added by WebVpnPortForward`
If any lines contain this string, your hosts file is clientless SSL VPN-customized. If your hosts file is clientless SSL VPN-customized, it looks similar to the following example:

```
123.0.0.3 server1 # added by WebVpnPortForward
123.0.0.3 server1.example.com vpn3000.com # added by WebVpnPortForward
123.0.0.4 server2 # added by WebVpnPortForward
123.0.0.4 server2.example.com.vpn3000.com # added by WebVpnPortForward
123.0.0.5 server3 # added by WebVpnPortForward
123.0.0.5 server3.example.com vpn3000.com # added by WebVpnPortForward

# Copyright (c) 1993-1999 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     cisco.example.com          # source server
#       38.25.63.10     x.example.com              # x client host

123.0.0.1       localhost
```

Step 3 Delete the lines that contain the string: `# added by WebVpnPortForward`

Step 4 Save and close the file.

Step 5 Start clientless SSL VPN and log in.

The home page appears.

Step 6 Click the Application Access link.

The Application Access window appears. Application Access is now enabled.

Configuring File Access

Clientless SSL VPN serves remote users with HTTPS portal pages that interface with proxy CIFS and/or FTP clients running on the security appliance. Using either CIFS or FTP, clientless SSL VPN provides users with network access to the files on the network, to the extent that the users meet user authentication requirements and the file properties do not restrict access. The CIFS and FTP clients are transparent; the portal pages delivered by clientless SSL VPN provide the appearance of direct access to the file systems.

When a user requests a list of files, clientless SSL VPN queries the server designated as the master browser for the IP address of the server containing the list. The security appliance gets the list and delivers it to the remote user on a portal page.

Clientless SSL VPN lets the user invoke the following CIFS and FTP functions, depending on user authentication requirements and file properties:

- Navigate and list domains and workgroups, servers within a domain or workgroup, shares within a server, and files within a share or directory
- Create directories
- Download, upload, rename, move, and delete files

The security appliance uses a master browser, WINS server, or DNS server, typically on the same network as the security appliance or reachable from that network, to query the network for a list of servers when the remote user clicks Browse Networks in the menu of the portal page or on the toolbar displayed during the Clientless SSL VPN session.

The master browser or DNS server provides the CIFS/FTP client on the security appliance with a list of the resources on the network, which clientless SSL VPN serves to the remote user.

Note Before configuring file access, you must configure the shares on the servers for user access.

Adding Support for File Access

Configure file access as follows:

Note Step 1 of this procedure describes how to specify the master browser and WINS servers. As an alternative, you can use ASDM to configure URL lists and entries that provide access to file shares.

Adding a share in ASDM does not require a master browser or a WINS server. However, it does not provide support for the Browse Networks link. You can use a hostname or an IP address to refer to ServerA when entering this command. If you use a hostname, the security appliance requires a DNS server to resolve it to an IP address.

Step 1 Use the **nbns-server** command in tunnel-group webvpn configuration mode once for each NetBIOS Name Server (NBNS). This step lets you browse a network or domain.

nbns-server {*IPaddress* | *hostname*} [**master**] [**timeout** *timeout*] [**retry** *retries*]

master is the computer designated as the master browser. The master browser maintains the list of computers and shared resources. Any NBNS server you identify with this command without entering the master portion of the command must be a Windows Internet Naming Server (WINS). Specify the master browser first, then specify the WINS servers. You can specify up to three servers, including the master browser, for a connection profile.

retries is the number of times to retry queries to the NBNS server. The security appliance recycles through the list of servers this number of times before sending an error message. The default value is 2; the range is 1 through 10.

timeout is the number of seconds the security appliance waits before sending the query again, to the same server if it is the only one, or another server if there are more than one. The default timeout is 2 seconds; the range is 1 to 30 seconds.

For example,

```
hostname(config-tunnel-webvpn)# nbns-server 192.168.1.20 master
hostname(config-tunnel-webvpn)# nbns-server 192.168.1.41
hostname(config-tunnel-webvpn)# nbns-server 192.168.1.47
```

Note Use the **show tunnel-group webvpn-attributes** command if you want to display the NBNS servers already present in the connection profile configuration.

Step 2 (Optional) Use the **character-encoding** command to specify the character set to encode in clientless SSL VPN portal pages to be delivered to remote users. By default, the encoding type set on the remote browser determines the character set for clientless SSL VPN portal pages, so you need to set the character encoding only if it is necessary to ensure proper encoding on the browser.

character-encoding *charset*

Charset is a string consisting of up to 40 characters, and equal to one of the valid character sets identified in http://www.iana.org/assignments/character-sets. You can use either the name or the alias of a character set listed on that page. Examples include iso-8859-1, shift_jis, and ibm850.

Note The character-encoding and file-encoding values do not exclude the font family to be used by the browser. You need to complement the setting of one these values with the **page style** command in webvpn customization command mode to replace the font family if you are using Japanese Shift_JIS character encoding, as shown in the following example, or enter the **no page style** command in webvpn customization command mode to remove the font family.

The following example sets the character-encoding attribute to support Japanese Shift_JIS characters, removes the font family, and retains the default background color:

```
hostname(config-webvpn)# character-encoding shift_jis
hostname(config-webvpn)# customization DfltCustomization
hostname(config-webvpn-custom)# page style background-color:white
```

Step 3 (Optional) Use the **file-encoding** command to specify the encoding for clientless SSL VPN portal pages from specific CIFS servers. Thus, you can use different file-encoding values for CIFS servers that require different character encodings.

file-encoding {server-name | server-ip-address} *charset*

The following example sets the file-encoding attribute of the CIFS server 10.86.5.174 to support IBM860 (alias “CP860”) characters:

```
hostname(config-webvpn)# file-encoding 10.86.5.174 cp860
```

For a complete description of these commands, see the *Cisco Security Appliance Command Reference*.

Ensuring Clock Accuracy for SharePoint Access

The clientless SSL VPN server on the security appliance uses cookies to interact with applications such as Microsoft Word on the endpoint. The cookie expiration time set by the security appliance can cause Word to malfunction when accessing documents on a SharePoint server if the time on the security appliance is incorrect. To prevent this malfunction, set the ASA clock properly. We recommend configuring the security appliance to dynamically synchronize the time with an NTP server. For instructions, see “Setting the Date and Time.”

Using Clientless SSL VPN with PDAs

You can access Clientless SSL VPN from your Pocket PC or other certified personal digital assistant device. Neither the security appliance administrator nor the Clientless SSL VPN user need do anything special to use Clientless SSL VPN with a certified PDA.

Cisco has certified the following PDA platform:

HP iPaq H4150
Pocket PC 2003
Windows CE 4.20.0, build 14053
Pocket Internet Explorer (PIE)
ROM version 1.10.03ENG
ROM Date: 7/16/2004

Some differences in the PDA version of Clientless SSL VPN exist:

- A banner web page replaces the popup Clientless SSL VPN window.
- An icon bar replaces the standard Clientless SSL VPN floating toolbar. This bar displays the Go, Home and Logout buttons.
- The Show Toolbar icon is not included on the main Clientless SSL VPN portal page.
- Upon Clientless SSL VPN logout, a warning message provides instructions for closing the PIE browser properly. If you do not follow these instructions and you close the browser window in the common way, PIE does not disconnect from Clientless SSL VPN or any secure website that uses HTTPS.
- Clientless SSL VPN supports OWA 2000 and OWA 2003 Basic Authentication. If Basic Authentication is not configured on an OWA server and a Clientless SSL VPN user attempts to access that server, access is denied.
- Unsupported Clientless SSL VPN features:
  - Application Access and other Java-dependent features.
  - HTTP proxy.
  - Cisco Secure Desktop provides limited support for Microsoft Windows CE.
  - Microsoft Outlook Web Access (OWA) 5.5.
  - The Citrix Metaframe feature (if the PDA does not have the corresponding Citrix ICA client software).

Using E-Mail over Clientless SSL VPN

Clientless SSL VPN supports several ways to access e-mail. This section includes the following methods:

- Configuring E-mail Proxies
- Configuring Web E-mail: MS Outlook Web Access

Configuring E-mail Proxies

Clientless SSL VPN supports IMAP4S, POP3S, and SMTPS e-mail proxies. Table 39-8 lists attributes that apply globally to e-mail proxy users:

Table 39-8 Attributes for E-mail Proxy Users over Clientless SSL VPN

Function	Command	Default Value
Specifies the previously configured accounting servers to use with e-mail proxy.	**accounting-server-group**	None
Specifies the authentication method(s) for e-mail proxy users.	**authentication**	IMAP4S: Mailhost (required) POP3S Mailhost (required) SMTPS: AAA
Specifies the previously configured authentication servers to use with e-mail proxy.	**authentication-server-group**	LOCAL
Specifies the previously configured authorization servers to use with Clientless SSL VPN.	**authorization-server-group**	None
Requires users to authorize successfully to connect.	**authorization-required**	Disabled
Identifies the DN of the peer certificate to use as a username for authorization.	**authorization-dn-attributes**	Primary attribute: CN Secondary attribute: OU
Specifies the name of the group policy to use.	**default-group-policy**	DfltGrpPolicy
Enables e-mail proxy on the specified interface.	**enable**	Disabled
Defines the separator between the e-mail and VPN usernames and passwords.	**name-separator**	":" (colon)
Configures the maximum number of outstanding non-authenticated sessions.	**outstanding**	20
Sets the port the e-mail proxy listens to.	**port**	IMAP4S:993 POP3S: 995 SMTPS: 988[1]
Specifies the default e-mail server.	**server**	None.
Defines the separator between the e-mail and server names.	**server-separator**	"@"

1. With the Eudora e-mail client, SMTPS works only on port 465, even though the default port for SMTPS connections is 988.

E-mail Proxy Certificate Authentication

E-mail clients such as MS Outlook, MS Outlook Express, and Eudora lack the ability to access the certificate store.

Configuring Web E-mail: MS Outlook Web Access

Web e-mail is MS Outlook Web Access for Exchange 2000, Exchange 5.5, and Exchange 2003. It requires an MS Outlook Exchange Server at the central site. It also requires that users perform the following tasks:

- Enter the URL of the mail server in a browser in your Clientless SSL VPN session.
- When prompted, enter the e-mail server username in the format *domain\username*.
- Enter the e-mail password.

Optimizing Clientless SSL VPN Performance

The security appliance provides several ways to optimize Clientless SSL VPN performance and functionality. Performance improvements include caching and compressing web objects. Functionality tuning includes setting limits on content transformation and proxy-bypass. APCF provides an additional method of tuning content transformation. The following sections explain these features:

- Configuring Caching
- Configuring Content Transformation

Configuring Caching

Caching enhances Clientless SSL VPN performance. It stores frequently reused objects in the system cache, which reduces the need to perform repeated rewriting and compressing of content. It reduces traffic between Clientless SSL VPN and the remote servers, with the result that many applications run much more efficiently.

By default, caching is enabled. You can customize the way caching works for your environment by using the caching commands in cache mode, which you enter from webvpn mode, as in the following example.

```
hostname(config)#
hostname(config)# webvpn
hostname(config-webvpn)# cache
```

A list of caching commands and their functions follows:

Cache Command	Function
disable	Disables caching.
expiry-time	Configures an expiration time for caching objects.
lmfactor	Configures terms for revalidating cached objects.
max-object-size	Sets a maximum size for objects to cache.
min-object-size	Sets a minimum size for objects to cache.
cache-static-content	Caches all cacheable web objects, content not subject to rewriting. Examples include images and PDF files.

Configuring Content Transformation

By default, the security appliance processes all Clientless SSL VPN traffic through a content transformation/rewriting engine that includes advanced elements such as JavaScript and Java to proxy HTTP traffic that may have different semantics and access control rules depending on whether the user is accessing an application within or independently of an SSL VPN device.

Some web resources require highly individualized treatment. The following sections describe functionality that provides such treatment:

- Configuring a Certificate for Signing Rewritten Java Content
- Disabling Content Rewrite
- Using Proxy Bypass
- Configuring Application Profile Customization Framework

Subject to the requirements of your organization and the web content involved, you might use one of these features.

Configuring a Certificate for Signing Rewritten Java Content

Java objects which have been transformed by Clientless SSL VPN can subsequently be signed using a PKCS12 digital certificate associated with a trustpoint. You import and employ the certificate using a combination of the **crypto ca import** and **java-trustpoint** commands.

The following example commands show the creation of a trustpoint named mytrustpoint and its assignment to signing Java objects:

```
hostname(config)# crypto ca import mytrustpoint pkcs12 mypassphrase
Enter the base 64 encoded PKCS12.
End with the word "quit" on a line by itself.
[ PKCS12 data omitted ]
quit
INFO: Import PKCS12 operation completed successfully.
hostname(config)# webvpn
hostname(config)# java-trustpoint mytrustpoint
```

Disabling Content Rewrite

You might not want some applications and web resources, for example, public websites, to go through the security appliance. The security appliance therefore lets you create rewrite rules that let users browse certain sites and applications without going through the security appliance. This is similar to split-tunneling in an IPSec VPN connection.

Use the **rewrite** command with the **disable** option in webvpn mode to specify applications and resources to access outside a Clientless SSL VPN tunnel.

You can use the rewrite command multiple times. The order number of rules is important because the security appliance searches rewrite rules by order number, starting with the lowest, and applies the first rule that matches.

Using Proxy Bypass

You can configure the security appliance to use proxy bypass when applications and web resources work better with the special content rewriting this feature provides. Proxy bypass is an alternative method of content rewriting that makes minimal changes to the original content. It is often useful with custom web applications.

You can use this command multiple times. The order in which you configure entries is unimportant. The interface and path mask or interface and port uniquely identify a proxy bypass rule.

If you configure proxy bypass using ports rather than path masks, depending on your network configuration, you might need to change your firewall configuration to allow these ports access to the security appliance. Use path masks to avoid this restriction. Be aware, however, that path masks can change, so you might need to use multiple pathmask statements to exhaust the possibilities.

A path is everything in a URL after the .com or .org or other types of domain name. For example, in the URL www.mycompany.com/hrbenefits, *hrbenefits* is the path. Similarly, for the URL www.mycompany.com/hrinsurance, *hrinsurance* is the path. If you want to use proxy bypass for all hr sites, you can avoid using the command multiple times by using the * wildcard as follows: /hr*.

To configure proxy bypass, use the **proxy-bypass** command in webvpn mode.

Configuring Application Profile Customization Framework

An APCF profile for Clientless SSL VPN lets the security appliance handle non-standard applications and web resources so that they display correctly over a Clientless SSL VPN connection. An APCF profile contains a script that specifies when (pre, post), where (header, body, request, response), and what data to transform for a particular application. The script is in XML and uses sed (stream editor) syntax for string/text transformation. Multiple APCF profiles can run in parallel on a security appliance. Within an APCF profile script, multiple APCF rules can apply. In this case, the security appliance processes the oldest rule first (based on configuration history), then the next oldest rule, and so forth.

You can store APCF profiles on the security appliance flash memory, or on an HTTP, HTTPS, or TFTP server. Use the **apcf** command in webvpn mode to identify and locate an APCF profile that you want to load on the security appliance.

Note We recommend that you configure an APCF profile only with the assistance of Cisco personnel.

The following example shows how to enable an APCF profile named apcf1.xml, located on flash memory.

```
hostname(config)# webvpn
hostname(config-webvpn)# apcf flash:/apcf/apcf1.xml
```

This example shows how to enable an APCF profile named apcf2.xml, located on an https server called myserver, port 1440 with the path being /apcf.

```
hostname(config)# webvpn
hostname(config-webvpn)# apcf https://myserver:1440/apcf/apcf2.xml
```

APCF Syntax

Caution Misuse of an APCF profile can result in reduced performance and undesired rendering of content. In most cases, Cisco Engineering supplies APCF profiles to solve specific application rendering issues.

APCF profiles use XML format, and sed script syntax, with the XML tags in Table 39-9.

Table 39-9 APCF XML Tags

Tag	Use
<APCF>...</APCF>	The mandatory root element that opens any APCF XML file.
<version>1.0</version>	The mandatory tag that specifies the APCF implementation version. Currently the only version is 1.0.
<application>...</application>	The mandatory tag that wraps the body of the XML description.
<id> text </id>	The mandatory tag that describes this particular APCF functionality.
<apcf-entities>...</apcf-entities>	The mandatory tag that wraps a single or multiple APCF entities.

Table 39-9 APCF XML Tags (continued)

Tag	Use
<js-object>…</js-object> <html-object>…</html-object> <process-request-header>...</preprocess-request-header> <process-response-header>...</preprocess-response-header> <preprocess-request-body>...</preprocess-request-body> <postprocess-request-body>...</postprocess-request-body> <preprocess-response-body>...</preprocess-response-body> <postprocess-response-body>...</postprocess-response-body>	One of theses tags specifying type of content or the stage at which the APCF processing should take place is required.
<conditions>… </conditions>	A child element of the pre/post-process tags that specifies criteria for processing such as: http-version (such as 1.1, 1.0, 0.9) http-method (get, put, post, webdav) http-scheme (http, https, other) server-regexp regular expression containing ("a".."z" | "A".."Z" | "0".."9" | ".-_*[]?")) server-fnmatch (regular expression containing ("a".."z" | "A".."Z" | "0".."9" | ".-_*[]?+()\{},")), user-agent-regexp user-agent-fnmatch request-uri-regexp request-uri-fnmatch If more than one of condition tags are present, the security appliance performs a logical AND for all tags.
<action> … </action>	Wraps one or more actions to perform on the content under specified conditions; define each of these actions with the following <do> tag or the <sed-script> tag.
<do>…</do>	Defines one of the following actions: <no-rewrite/> <no-toolbar/> <no-gzip/> <force-cache/> <force-no-cache/>
<sed-script> TEXT </sed-script>	The child element of the action tag. The TEXT must be a valid Sed script. The <sed-script> applies to the <conditions> tag defined before it.

APCF Example

The following example shows what an APCF profile looks like.

```
<APCF>
<version>1.0</version>
<application>
  <id>Do not compress content from notsogood.com</id>
  <apcf-entities>
      <process-request-header>
         <conditions>
           <server-fnmatch>*.notsogood.com</server-fnmatch>
         </conditions>
           <action>
             <do><no-gzip/></do>
           </action>
      </process-request-header>
  </apcf-entities>
</application>
</APCF>
```

Clientless SSL VPN End User Setup

This section is for the system administrator who sets up Clientless SSL VPN for end users. It describes how to customize the end-user interface.

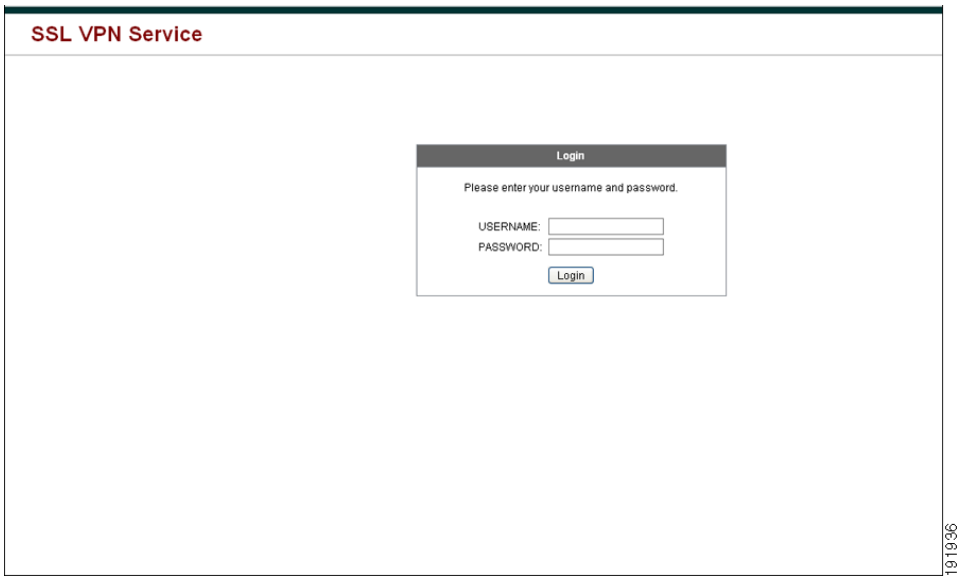
This section summarizes configuration requirements and tasks for a remote system. It specifies information to communicate to users to get them started using Clientless SSL VPN. It includes the following topics:

- Defining the End User Interface
- Customizing Clientless SSL VPN Pages, page 39-59
- Customizing Help, page 39-71
- Requiring Usernames and Passwords
- Communicating Security Tips
- Configuring Remote Systems to Use Clientless SSL VPN Features
- Translating the Language of User Messages

Defining the End User Interface

The Clientless SSL VPN end user interface consists of a series of HTML panels. A user logs on to Clientless SSL VPN by entering the IP address of a security appliance interface in the format https://*address*. The first panel that displays is the login screen (Figure 39-6).

Figure 39-6 Clientless SSL VPN Login Screen



Viewing the Clientless SSL VPN Home Page

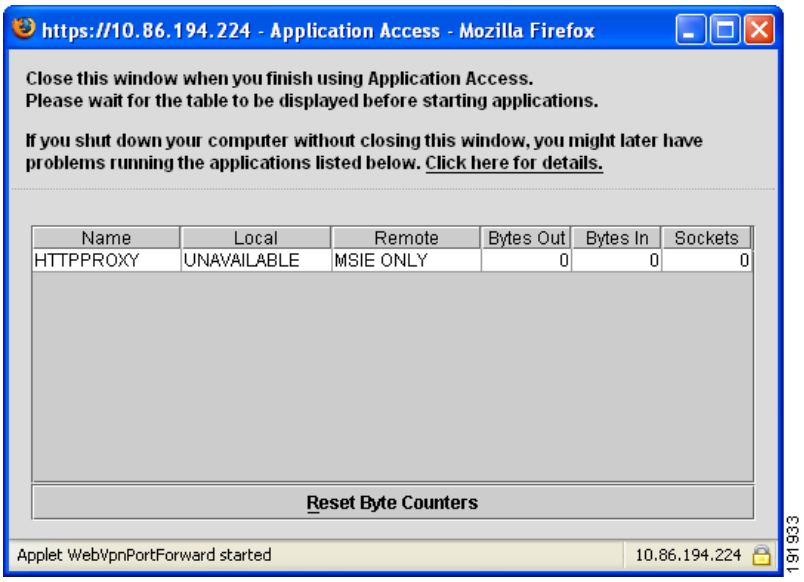
After the user logs in, the portal page opens.

The home page displays all of the Clientless SSL VPN features you have configured, and its appearance reflects the logo, text, and colors you have selected. This sample home page includes all available Clientless SSL VPN features with the exception of identifying specific file shares. It lets users browse the network, enter URLs, access specific websites, and use Application Access (port forwarding and smart tunnels) to access TCP applications.

Viewing the Clientless SSL VPN Application Access Panel

To start port forwarding or smart tunnels, a user clicks the Go button in the Application Access box. The Application Access window opens (Figure 39-7).

Figure 39-7 Clientless SSL VPN Application Access Window



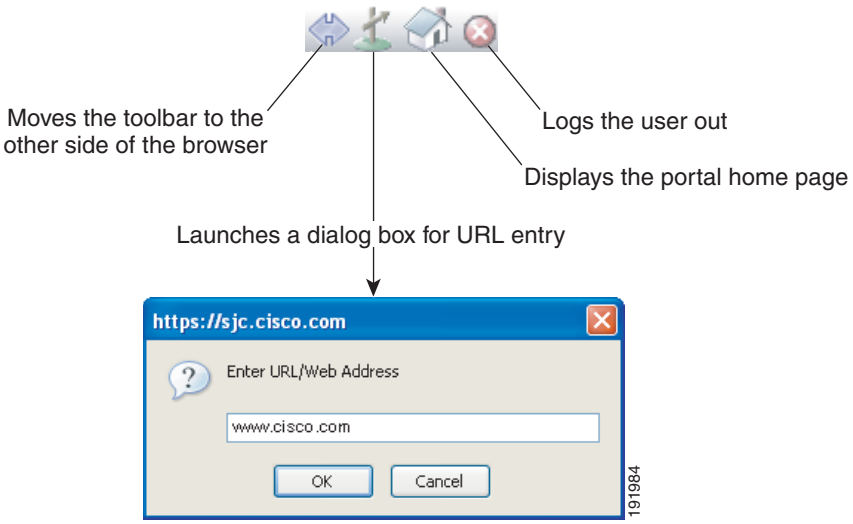
This window displays the TCP applications configured for this Clientless SSL VPN connection. To use an application with this panel open, the user starts the application in the normal way.

Note A stateful failover does not retain sessions established using Application Access. Users must reconnect following a failover.

Viewing the Floating Toolbar

The floating toolbar shown in Figure 39-8 represents the current Clientless SSL VPN session.

Figure 39-8 Clientless SSL VPN Floating Toolbar



Be aware of the following characteristics of the floating toolbar:

- The toolbar lets you enter URLs, browse file locations, and choose preconfigured web connections without interfering with the main browser window.
- If you configure your browser to block popups, the floating toolbar cannot display.
- If you close the toolbar, the security appliance prompts you to confirm that you want to end the Clientless SSL VPN session.

See Table 39-12 on page 39-76 for detailed information about using Clientless SSL VPN.

Customizing Clientless SSL VPN Pages

You can change the appearance of the portal pages displayed to Clientless SSL VPN users. This includes the Login page displayed to users when they connect to the security appliance, the Home page displayed to users after the security appliance authenticates them, the Application Access window displayed when users launch an application, and the Logout page displayed when users logout of Clientless SSL VPN sessions.

After you customize the portal pages, you can save your customization and apply it to a specific connection profile, group policy, or user. You can create and save many customization objects, enabling the security appliance to change the appearance of portal pages for individual users or groups of users.

This section contains the following topics and tasks:

- How Customization Works, page 39-59
- Exporting a Customization Template, page 39-60
- Editing the Customization Template, page 39-60
- Importing a Customization Object, page 39-66
- Applying Customizations to Connection Profiles, Group Policies and Users, page 39-66
- Login Screen Advanced Customization, page 39-67

How Customization Works

The security appliance uses customization objects to define the appearance of user screens. A customization object is compiled from an XML file which contains XML tags for all the customizable screen items displayed to remote users. The security appliance software contains a customization template that you can export to a remote PC. You can edit this template and import the template back into the security appliance as a new customization object.

When you export a customization object, an XML file containing XML tags is created at the URL you specify. The XML file created by the customization object named *Template* contains empty XML tags, and provides the basis for creating new customization objects. This object cannot be changed or deleted from cache memory, but can be exported, edited, and imported back into the security appliance as a new customization object.

Customization Objects, Connection Profiles, and Group Policies

Initially, when a user first connects, the default customization object (named *DfltCustomization*) identified in the connection profile (tunnel group) determines how the logon screen appears. If the connection profile list is enabled, and the user selects a different group, and that group has its own customization, the screen changes to reflect the customization object for that new group.

After the remote user is authenticated, the screen appearance is determined by whether a customization object that has been assigned to the group policy.

Exporting a Customization Template

When you export a customization object, an XML file is created at the URL you specify. The customization template (named *Template*) contains empty XML tags, and provides the basis for creating new customization objects. This object cannot be changed or deleted from cache memory, but can be exported, edited, and imported back into the security appliance as a new customization object.

You can export a customization object using the **export webvpn customization** command, make changes to the XML tags, and import the file as a new object using the **import webvpn customization** command.

The following example exports the default customization object (DfltCustomization) and creates the XML file named *dflt_custom*:

```
hostname# export webvpn customization DfltCustomization tftp://209.165.200.225/dflt_custom
!!!!!!!!!!!!!!!!INFO: Customization object 'DfltCustomization' was exported to
tftp://10.86.240.197/dflt_custom
hostname#
```

Editing the Customization Template

This section shows the contents of the customization template and has convenient figures to help you quickly choose the correct XML tag and make changes that affect the screens.

You can use a text editor or an XML editor to edit the XML file. The following example shows the XML tags of the customization template. Some redundant tags have been removed for easier viewing:

```
<custom>
   <localization>
      <languages>en,ja,zh,ru,ua</languages>
      <default-language>en</default-language>
   </localization>
    <auth-page>
      <window>
          <title-text l10n="yes"><![CDATA[SSL VPN Service]]></title-text>
      </window>
      <full-customization>
          <mode>disable</mode>
          <url></url>
      </full-customization>
      <language-selector>
        <mode>disable</mode>
        <title l10n="yes">Language:</title>
        <language>
          <code>en</code>
          <text>English</text>
        </language>
        <language>
          <code>zh</code>
          <text>ä¸-å›½ (Chinese)</text>
        </language>
        <language>
          <code>ja</code>
          <text>æ—¥æœ¬ (Japanese)</text>
        </language>
        <language>
          <code>ru</code>
          <text>Ð ÑƒÑÑкий (Russian)</text>
```

```
        </language>
        <language>
          <code>ua</code>
          <text>Ð£ÐºÑ Ð°Ñ–Ð½ÑÑŒÐºÐ° (Ukrainian)</text>
        </language>
      </language-selector>
      <logon-form>
          <title-text l10n="yes"><![CDATA[Login]]></title-text>
          <title-background-color><![CDATA[#666666]]></title-background-color>
          <title-font-color><![CDATA[#ffffff]]></title-font-color>
          <message-text l10n="yes"><![CDATA[Please enter your username and
password.]]></message-text>
          <username-prompt-text l10n="yes"><![CDATA[USERNAME:]]></username-prompt-text>
          <password-prompt-text l10n="yes"><![CDATA[PASSWORD:]]></password-prompt-text>
          <internal-password-prompt-text l10n="yes">Internal
Password:</internal-password-prompt-text>
          <internal-password-first>no</internal-password-first>
          <group-prompt-text l10n="yes"><![CDATA[GROUP:]]></group-prompt-text>
          <submit-button-text l10n="yes"><![CDATA[Login]]></submit-button-text>
          <title-font-color><![CDATA[#ffffff]]></title-font-color>
          <title-background-color><![CDATA[#666666]]></title-background-color>
          <font-color>#000000</font-color>
          <background-color>#ffffff</background-color>
          <border-color>#858A91</border-color>
      </logon-form>
      <logout-form>
          <title-text l10n="yes"><![CDATA[Logout]]></title-text>
          <message-text l10n="yes"><![CDATA[Goodbye.<br>

For your own security, please:<br>

<li>Clear the browser's cache

<li>Delete any downloaded files

<li>Close the browser's window]]></message-text>
          <login-button-text l10n="yes">Logon</login-button-text>
          <hide-login-button>no</hide-login-button>
          <title-background-color><![CDATA[#666666]]></title-background-color>
          <title-font-color><![CDATA[#ffffff]]></title-font-color>
          <title-font-color><![CDATA[#ffffff]]></title-font-color>
          <title-background-color><![CDATA[#666666]]></title-background-color>
          <font-color>#000000</font-color>
          <background-color>#ffffff</background-color>
          <border-color>#858A91</border-color>
      </logout-form>
      <title-panel>
         <mode>enable</mode>
         <text l10n="yes"><![CDATA[SSL VPN Service]]></text>
         <logo-url l10n="yes">/+CSCOU+/csco_logo.gif</logo-url>
         <gradient>yes</gradient>
         <style></style>
         <background-color><![CDATA[#ffffff]]></background-color>
         <font-size><![CDATA[larger]]></font-size>
         <font-color><![CDATA[#800000]]></font-color>
         <font-weight><![CDATA[bold]]></font-weight>
      </title-panel>
      <info-panel>
        <mode>disable</mode>
        <image-url l10n="yes">/+CSCOU+/clear.gif</image-url>
        <image-position>above</image-position>
        <text l10n="yes"></text>
      </info-panel>
      <copyright-panel>
```

```
      <mode>disable</mode>
      <text l10n="yes"></text>
    </copyright-panel>
  </auth-page>
  <portal>
    <title-panel>
      <mode>enable</mode>
      <text l10n="yes"><![CDATA[SSL VPN Service]]></text>
      <logo-url l10n="yes">/+CSCOU+/csco_logo.gif</logo-url>
      <gradient>yes</gradient>
      <style></style>
      <background-color><![CDATA[#ffffff]]></background-color>
      <font-size><![CDATA[larger]]></font-size>
      <font-color><![CDATA[#800000]]></font-color>
      <font-weight><![CDATA[bold]]></font-weight>
    </title-panel>
    <browse-network-title l10n="yes">Browse Entire Network</browse-network-title>
    <access-network-title l10n="yes">Start AnyConnect</access-network-title>
    <application>
      <mode>enable</mode>
      <id>home</id>
      <tab-title l10n="yes">Home</tab-title>
      <order>1</order>
    </application>
    <application>
      <mode>enable</mode>
      <id>web-access</id>
      <tab-title l10n="yes"><![CDATA[Web Applications]]></tab-title>
      <url-list-title l10n="yes"><![CDATA[Web Bookmarks]]></url-list-title>
      <order>2</order>
    </application>
    <application>
      <mode>enable</mode>
      <id>file-access</id>
      <tab-title l10n="yes"><![CDATA[Browse Networks]]></tab-title>
      <url-list-title l10n="yes"><![CDATA[File Folder Bookmarks]]></url-list-title>
      <order>3</order>
    </application>
    <application>
      <mode>enable</mode>
      <id>app-access</id>
      <tab-title l10n="yes"><![CDATA[Application Access]]></tab-title>
      <order>4</order>
    </application>
    <application>
      <mode>enable</mode>
      <id>net-access</id>
      <tab-title l10n="yes">AnyConnect</tab-title>
      <order>4</order>
    </application>
    <application>
      <mode>enable</mode>
      <id>help</id>
      <tab-title l10n="yes">Help</tab-title>
      <order>1000000</order>
    </application>
    <toolbar>
      <mode>enable</mode>
      <logout-prompt-text l10n="yes">Logout</logout-prompt-text>
      <prompt-box-title l10n="yes">Address</prompt-box-title>
      <browse-button-text l10n="yes">Browse</browse-button-text>
    </toolbar>
    <column>
      <width>100%</width>
```

```
            <order>1</order>
         </column>
         <pane>
            <type>TEXT</type>
            <mode>disable</mode>
            <title></title>
            <text></text>
            <notitle></notitle>
            <column></column>
            <row></row>
            <height></height>
         </pane>
         <pane>
            <type>IMAGE</type>
            <mode>disable</mode>
            <title></title>
            <url l10n="yes"></url>
            <notitle></notitle>
            <column></column>
            <row></row>
            <height></height>
         </pane>
         <pane>
            <type>HTML</type>
            <mode>disable</mode>
            <title></title>
            <url l10n="yes"></url>
            <notitle></notitle>
            <column></column>
            <row></row>
            <height></height>
         </pane>
         <pane>
            <type>RSS</type>
            <mode>disable</mode>
            <title></title>
            <url l10n="yes"></url>
            <notitle></notitle>
            <column></column>
            <row></row>
            <height></height>
         </pane>
         <url-lists>
            <mode>group</mode>
         </url-lists>
         <home-page>
            <mode>standard</mode>
            <url></url>
         </home-page>
      </portal>
</custom>
```

Figure 39-9 shows the Logon page and its customizing XML tags. All these tags are nested within the higher-level tag <auth-page>.

Figure 39-9 Logon Page and Associated XML Tags

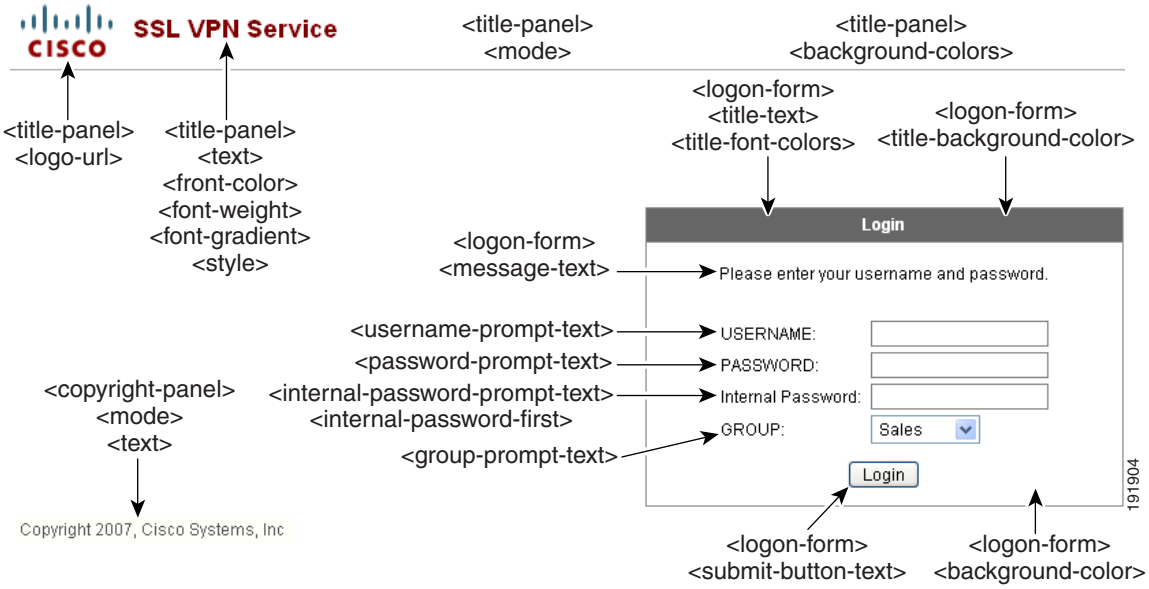


Figure 39-10 shows the Language Selector drop-down list that is available on the Logon page, and the XML tags for customizing this feature. All these tags are nested within the higher-level <auth-page> tag.

Figure 39-10 Language Selector on Logon Screen and Associated XML Tags

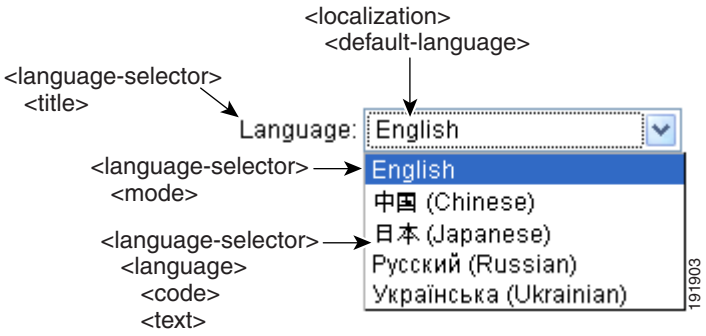


Figure 39-11 shows the Information Panel that is available on the Logon page, and the XML tags for customizing this feature. This information can appear to the left or right of the login box. These tags are nested within the higher-level <auth-page> tag.

Figure 39-11 Information Panel on Logon Screen and Associated XML Tags

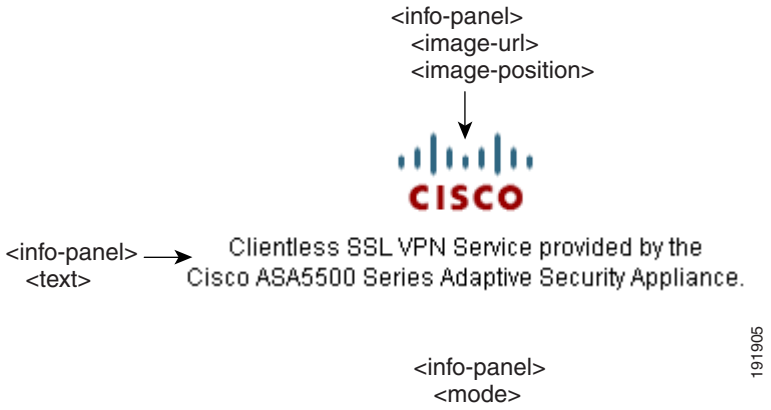
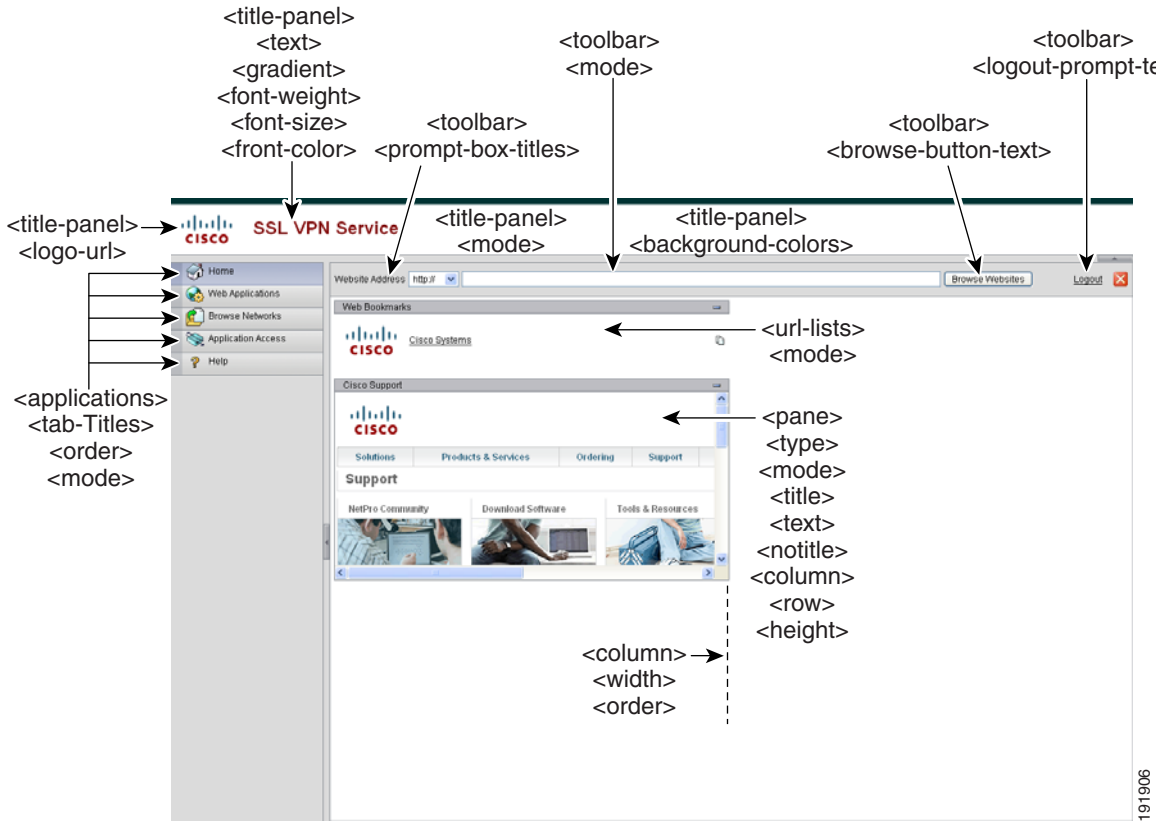


Figure 39-12 shows the Portal page and the XML tags for customizing this feature. These tags are nested within the higher-level <auth-page> tag.

Figure 39-12 Portal Page and Associated XML Tags



Importing a Customization Object

After you edit and save the XML file, import it into cache memory of the security appliance using the **import webvpn customization** command from EXEC mode. When you import the customization object, the security appliance checks the XML code for validity. If the code is valid, the security appliance stores the object in a hidden location in cache memory.

The following example imports the customization object *General.xml* from the URL 209.165.201.22/customization and names it *custom1*.

```
hostname# import webvpn customization custom1 tftp://209.165.201.22/customization
/General.xml
Accessing
tftp://209.165.201.22/customization/General.xml...!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Writing file disk0:/csco_config/97/custom1...
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
329994 bytes copied in 5.350 secs (65998 bytes/sec)
```

Applying Customizations to Connection Profiles, Group Policies and Users

After you create a customization, you can apply the customization to a connection profile, a group, or a user, with the **customization** command. The options displayed with this command are different depending on the mode you are in.

Note Connection profiles were previously referred to as tunnel groups.

For more information about configuring connection profiles, group policies, and users, see Chapter 32, "Configuring Connection Profiles, Group Policies, and Users.".

Applying Customizations to Connection Profiles

To apply a customization to a connection profile, use the **customization** command from tunnel-group webvpn mode:

[**no**] **customization** *name*

name is the name of a customization to apply to the connection profile.

To remove the command from the configuration, and remove a customization from the connection profile, use the **no** form of the command.

Enter the **customization command followed by a question mark (?)** to view a list of existing customizations.

In the following example, the user enters tunnel-group webvpn mode and enables the customization *cisco* for the connection profile *cisco_telecommuters*:

```
hostname(config)# tunnel-group cisco_telecommuters webvpn-attributes
hostname(tunnel-group-webvpn)# customization cisco
```

Applying Customizations to Groups and Users

To apply a customization to a group or user, use the **customization** command from group policy webvpn mode or username webvpn mode. In these modes, the **none** and **value** options are included:

[**no**] **customization {none | value** *name*}

none disables the customization for the group or user, prevents the value from being inherited, and displays the default Clientless SSL VPN pages.

value *name* is the name of a customization to apply to the group or user.

To remove the command from the configuration, and cause the value to be inherited, use the **no** form of the command.

Enter the **customization value command followed by a question mark (?)** to view a list of existing customizations.

In the following example, the user enters group policy webvpn mode, queries the security appliance for a list of customizations, and enables the customization *cisco* for the group policy *cisco_sales*:

```
hostname(config)# group-policy cisco_sales attributes
hostname(config-group-policy)# webvpn
hostname(config-username-webvpn)# customization value ?

config-username-webvpn mode commands/options:
Available configured customization profiles:
    DfltCustomization
    cisco
hostname(config-group-webvpn)# customization value cisco
```

In the next example, the user enters username webvpn mode and enables the customization *cisco* for the user *cisco_employee*:

```
hostname(config)# username cisco_employee attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# customization value cisco
```

Login Screen Advanced Customization

If you prefer to use your own, custom login screen, rather than changing specific screen elements of the login screen we provide, you can perform this advanced customization using the *Full Customization* feature.

With Full Customization, you provide the HTML for your own login screen, and you insert Cisco HTML code that calls functions on the security appliance that create the Login form and the Language Selector drop-down list.

This section describes the modifications you need to make to your HTML code and the tasks required to configure the security appliance to use your code.

Figure 39-13 shows the standard Cisco login screen that displays to Clientless SSL VPN users. The Login form is displayed by a function called by the HTML code.

Figure 39-13 Standard Cisco Login Page

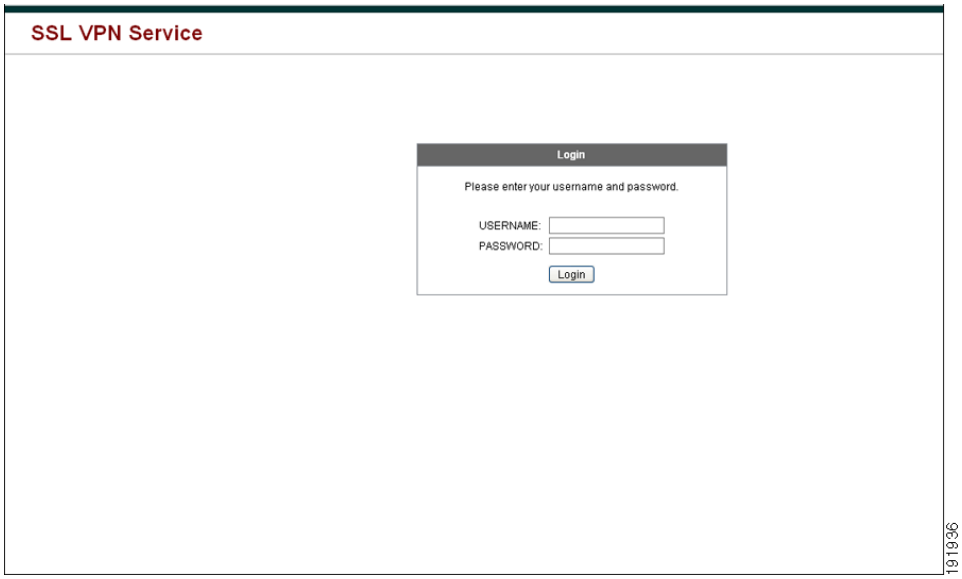


Figure 39-14 shows the Language Selector drop-down list. This feature is an option for Clientless SSL VPN users, and is also called by a function in the HTML code of the login screen.

Figure 39-14 Language Selector Drop-down List

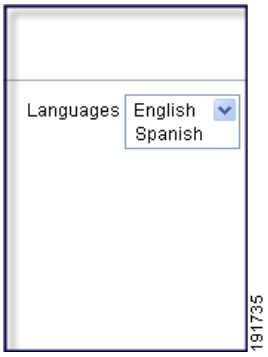
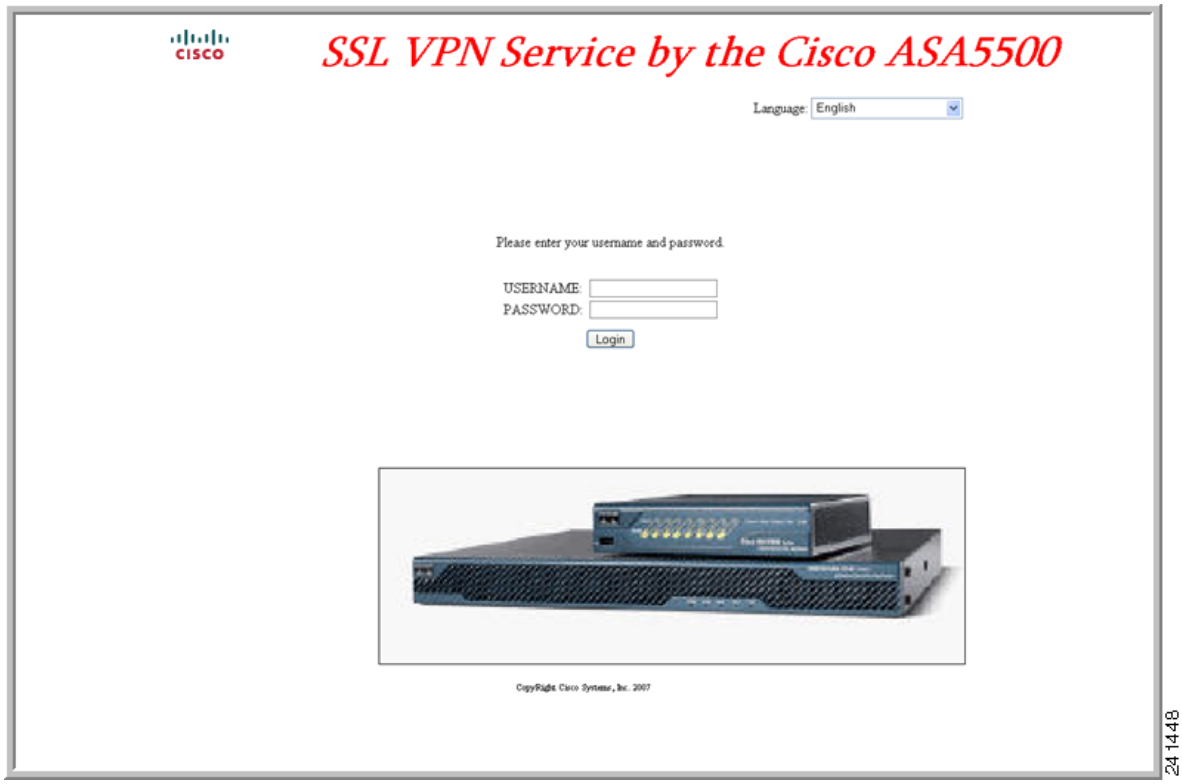


Figure 39-15 shows a simple example of a custom login screen enabled by the Full Customization feature.

Figure 39-15 Example of Full Customization of Login Screen



Example HTML Code for Custom Login Screen File

The following HTML code is used as an example and is the code that displays the screen shown in Figure 39-15:

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>New Page 3</title>
<base target="_self">
</head>

<p align="center">
<img border="0" src="/+CSCOU+/cisco_logo.jpg" width="188" height="48"><font face="Snap
ITC" size="6" color="#FF00FF">
</font><font face="Snap ITC" color="#FF00FF" size="7"> </font><i><b><font
color="#FF0000" size="7" face="Sylfaen"> SSL VPN Service by the Cisco
ASA5500</font></b></i></p>

    <body onload="csco_ShowLoginForm('lform');csco_ShowLanguageSelector('selector')">

    <table>

    <tr><td colspan=3 height=20 align=right><div id="selector" style="width:
    300px"></div></td></tr>
    <tr><td></td><td></td><td></td></tr>
    <tr>
    <td height="379"></td>
    <td height="379"></td>
    <td align=middle valign=middle>
    <div id=lform >
```

```
<p> </p>
<p> </p>
<p> </p>
<p>Loading...</p>
</div>
</td>
</tr>
<tr>
<td width="251"></td>
<td width="1"></td>
<td align=right valign=right width="800">
<img border="1" src="/+CSCOU+/asa5500.jpg" width="660" height="220" align="middle">
</td></tr>

</table>
```

The indented code injects the Login form and the Language Selector on the screen. The function **csco_ShowLoginForm('lform')** injects the logon form. **csco_ShowLanguageSelector('selector')** injects the Language Selector.

Full Customization Procedure

Follow these steps to modify your HTML file and configure the security appliance to use the new file:

Step 1 Name your file **logon.inc**. When you import the file, the security appliance recognizes this filename as the logon screen.

Step 2 Modify the paths of images used by the file to include **/+CSCOU+/**.

Files that are displayed to remote users before authentication must reside in a specific area of the security appliance cache memory represented by the path /+CSCOU+/. Therefore, the source for each image in the file must include this path. For example:

```
src="/+CSCOU+/asa5520.gif"
```

Step 3 Insert the special HTML code below. This code contains the Cisco functions, described earlier, that inject the login form and language selector onto the screen.

```
<body onload="csco_ShowLoginForm('lform');csco_ShowLanguageSelector('selector')">

<table>

<tr><td colspan=3 height=20 align=right><div id="selector" style="width:
300px"></div></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr>
<td height="379"></td>
<td height="379"></td>
<td align=middle valign=middle>
<div id=lform >
<p> </p>
<p> </p>
<p> </p>
<p>Loading...</p>
</div>
</td>
</tr>
<tr>
<td width="251"></td>
<td width="1"></td>
<td align=right valign=right width="800">
<img border="1" src="/+CSCOU+/asa5500.jpg" width="660" height="220" align="middle">
```

```
</td></tr>

</table>
```

Step 4 Import the file and images as Web Content using the **import webvpn webcontent** command from Privileged EXEC mode. For example:

```
hostname# import webvpn webcontent /+CSCOU+/login.inc tftp://209.165.200.225/login.inc
!!!!* Web resource `+CSCOU+/login.inc' was successfully initialized
hostname#
```

Step 5 Enable Full Customization in a customization object. First, export a customization template with the **export webvpn customization** command. For example:

```
hostname2# export webvpn customization template tftp://209.165.200.225/sales_vpn_login
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%INFO: Customization object 'Template' was exported to tftp://10.21.50.120/sales
_vpn_login
```

Then change the full customization mode tag in the file to enable, and supply the URL of the login file stored in the security appliance memory. For example:

```
<full-customization>
          <mode>enable</mode>
          <url>+CSCOU+/login.inc</url>
      </full-customization>
```

Now import the file as a new customization object. For example:

```
hostname# import webvpn customization sales_vpn_login tftp://10.21.50.120/sales_vpn_login$
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%INFO: customization object 'sales_vpn_login' was successfully imported
```

Step 6 Apply the customization object to a Connection Profile (tunnel group). For example:

```
hostname(config)# tunnel-group Sales webvpn-attributes
hostname(config-tunnel-webvpn)#customization sales_vpn_login
```

Customizing Help

The security appliance displays help content on the application panels during clientless SSL VPN sessions. You can customize the help files provided by Cisco or create help files in other languages. You then import them to flash memory for display during subsequent clientless sessions. You can also retrieve previously imported help content files, modify them, and reimport them to flash memory.

Each clientless application panel displays its own help file content using a predetermined filename. The prospective location of each is in the /+CSCOE+/help/*language*/ URL within flash memory of the security appliance. Table 39-10 shows the details about each of the help files you can maintain for clientless SSL VPN sessions.

Table 39-10 Clientless SSL VPN Application Help Files

Application Type	Panel	URL of Help File in Flash Memory of the Security Appliance	Help File Provided By Cisco in English?
Standard	Application Access	/+CSCOE+/help/*language*/app-access-hlp.inc	Yes
Standard	Browse Networks	/+CSCOE+/help/*language*/file-access-hlp.inc	Yes
Standard	AnyConnect Client	/+CSCOE+/help/*language*/net-access-hlp.inc	Yes
Standard	Web Access	/+CSCOE+/help/*language*/web-access-hlp.inc	Yes
Plug-in	MetaFrame Access	/+CSCOE+/help/*language*/ica-hlp.inc	No
Plug-in	Terminal Servers	/+CSCOE+/help/*language*/rdp-hlp.inc	Yes
Plug-in	Telnet/SSH Servers	/+CSCOE+/help/*language*/ssh,telnet-hlp.inc	Yes
Plug-in	VNC Connections	/+CSCOE+/help/*language*/vnc-hlp.inc	Yes

language is the abbreviation of the language rendered by the browser. This field is *not* used for file translation; it indicates the language used in the file. To specify a particular language code, copy the language abbreviation from the list of languages rendered by your browser. For example, a dialog window displays the languages and associated language codes when you use one of the following procedures:

- Open Internet Explorer and choose **Tools > Internet Options > Languages > Add**.
- Open Mozilla Firefox and choose **Tools > Options > Advanced > General**, click **Choose** next to Languages, and click **Select a language to add**.

The following sections describe how to customize the help content visible on clientless sessions:

- Customizing a Help File Provided By Cisco, page 39-72
- Creating Help Files for Languages Not Provided by Cisco, page 39-73
- Importing a Help File to Flash Memory, page 39-73
- Exporting a Previously Imported Help File from Flash Memory, page 39-74

Customizing a Help File Provided By Cisco

To customize a help file provided by Cisco, you need to get a copy of the file from the flash memory card first. Get the copy and customize it as follows:

Step 1 Use your browser to establish a clientless SSL VPN session with the security appliance.

Step 2 Display the help file by appending the string in "URL of Help File in Flash Memory of the Security Appliance" in Table 39-10, to the address of the security appliance, then press Enter.

Note Enter **en** in place of *language* to get the help file in English.

The following example address displays the English version of the Terminal Servers help:

https://*address_of_security_appliance***/+CSCOE+/help/en/rdp-hlp.inc**

Step 3 Choose File > Save (Page) As.

Caution Do not change the contents of the File name box.

Step 4 Change the Save as type option to "Web Page, HTML only" and click Save.

Step 5 Use your preferred HTML editor to modify the file.

Note You can use most HTML tags, but do *not* use tags that define the document and its structure (e.g., do not use <html>, <title>, <body>, <head>, <h1>, <h2>, etc. You can use character tags, such as the <b> tag, and the <p>, <ol>, <ul>, and <li> tags to structure content.

Step 6 Save the file as HTML only, using the original filename and extension.

Step 7 Make sure the filename matches the one in Table 39-10, and that it does not have an extra filename extension.

See "Importing a Help File to Flash Memory" to import the modified file for display in clientless SSL VPN sessions.

Creating Help Files for Languages Not Provided by Cisco

Use HTML to create help files in other languages.

Note You can use most HTML tags, but do *not* use tags that define the document and its structure (e.g., do not use <html>, <title>, <body>, <head>, <h1>, <h2>, etc. You can use character tags, such as the <b> tag, and the <p>, <ol>, <ul>, and <li> tags to structure content.

We recommend creating a separate folder for each language you want to support.

Save the file as HTML only. Use the filename following the last slash in "URL of Help File in Flash Memory of the Security Appliance" in Table 39-10.

See the next section to import the files for display in clientless SSL VPN sessions.

Importing a Help File to Flash Memory

To import a help content file to flash memory for display in clientless SSL VPN sessions, enter the following command in Privileged EXEC mode:

import webvpn webcontent *destination_url source_url*

destination_url is the string in "URL of Help File in Flash Memory of the Security Appliance" in Table 39-10.

source_url is the URL of the file to import. Valid prefixes are ftp://, http://, and tftp://.

The following example command copies the help file *app-access-hlp.inc* to flash memory from the TFTP server at 209.165.200.225. The URL includes the abbreviation *en* for the English language.

```
hostname# import webvpn webcontent /+CSCOE+/help/en/app-access-hlp.inc
tftp://209.165.200.225/app-access-hlp.inc
```

Exporting a Previously Imported Help File from Flash Memory

To retrieve a previously imported help content file for subsequent edits, enter the following command in Privileged EXEC mode:

export webvpn webcontent *source_url destination_url*

source_url is the string in "URL of Help File in Flash Memory of the Security Appliance" in Table 39-10.

destination_url is **the target URL**. Valid prefixes are ftp:// and tftp://. The maximum number of characters is 255.

The following example command copies the English language help file *file-access-hlp.inc* displayed on the Browse Networks panel to TFTP Server 209.165.200.225:

```
hostname# export webvpn webcontent /+CSCOE+/help/en/file-access-hlp.inc
tftp://209.165.200.225/file-access-hlp.inc
```

Requiring Usernames and Passwords

Depending on your network, during a remote session users might have to log in to any or all of the following: the computer itself, an Internet service provider, clientless SSL VPN, mail or file servers, or corporate applications. Users might have to authenticate in many different contexts, requiring different information, such as a unique username, password, or PIN.

Table 39-11 lists the type of usernames and passwords that clientless SSL VPN users might need to know.

Table 39-11 Usernames and Passwords to Give to Users of Clientless SSL VPN Sessions

Login Username/ Password Type	Purpose	Entered When
Computer	Access the computer	Starting the computer
Internet Service Provider	Access the Internet	Connecting to an Internet service provider
Clientless SSL VPN	Access remote network	Starting clientless SSL VPN
File Server	Access remote file server	Using the clientless SSL VPN file browsing feature to access a remote file server
Corporate Application Login	Access firewall-protected internal server	Using the clientless SSL VPN web browsing feature to access an internal protected website
Mail Server	Access remote mail server via Clientless SSL VPN	Sending or receiving e-mail messages

Communicating Security Tips

Advise users to always click the logout icon on the toolbar to close the clientless SSL VPN session. (Closing the browser window does not close the session.)

Advise users that using clientless SSL VPN does not ensure that communication with every site is secure. Clientless SSL VPN ensures the security of data transmission between the remote PC or workstation and the security appliance on the corporate network. If a user then accesses a non-HTTPS web resource (located on the Internet or on the internal network), the communication from the corporate security appliance to the destination web server is not private because it is not encrypted.

"Observing Clientless SSL VPN Security Precautions" on page 2 addresses an additional tip to communicate with users, depending on the steps you follow within that section.

Configuring Remote Systems to Use Clientless SSL VPN Features

Table 39-12 includes the following information about setting up remote systems to use clientless SSL VPN:

- Starting clientless SSL VPN
- Using the clientless SSL VPN Floating Toolbar
- Web Browsing
- Network Browsing and File Management
- Using Applications (Port Forwarding)
- Using E-mail via Port Forwarding
- Using E-mail via Web Access
- Using E-mail via e-mail proxy

Table 39-12 also provides information about the following:

- Clientless SSL VPN requirements, by feature
- Applications supported by clientless SSL VPN
- Client application installation and configuration requirements
- Information you might need to provide end users
- Tips and use suggestions for end users

It is possible you have configured user accounts differently and that different clientless SSL VPN features are available to each user. Table 39-12 organizes information by feature, so you can skip over the information for unavailable features.

Table 39-12 Remote System Configuration and End User Requirements for Clientless SSL VPN

Task	Remote System or End User Requirements	Specifications or Use Suggestions
Starting clientless SSL VPN	Connection to the Internet	Any Internet connection is supported, including: • Home DSL, cable, or dial-ups • Public kiosks • Hotel hook-ups • Airport wireless nodes • Internet cafes
	Web browsers supported by clientless SSL VPN	See the *Cisco ASA 5500 Series VPN Compatibility Reference*
	Cookies enabled on browser	Cookies must be enabled on the browser in order to access applications via port forwarding.
	URL for clientless SSL VPN	An https address in the following form: https://*address* where *address* is the IP address or DNS hostname of an interface of the security appliance (or load balancing cluster) on which clientless SSL VPN is enabled. For example: https://10.89.192.163 or https://cisco.example.com.
	Clientless SSL VPN username and password	—
	[Optional] Local printer	Clientless SSL VPN does not support printing from a web browser to a network printer. Printing to a local printer is supported.
Using the Floating Toolbar Displayed During a Clientless SSL VPN Session		A floating toolbar is available to simplify the use of clientless SSL VPN. The toolbar lets you enter URLs, browse file locations, and choose preconfigured web connections without interfering with the main browser window. If you configure your browser to block popups, the floating toolbar cannot display. The floating toolbar represents the current clientless SSL VPN session. If you click the **Close** button, the security appliance prompts you to confirm that you want to close the clientless SSL VPN session. **Tip** TIP: To paste text into a text field, use Ctrl-V. (Right-clicking is disabled on the toolbar displayed during the clientless SSL VPN session.)

Table 39-12 Remote System Configuration and End User Requirements for Clientless SSL VPN (continued)

Task	Remote System or End User Requirements	Specifications or Use Suggestions
Web Browsing	Usernames and passwords for protected websites	Using clientless SSL VPN does not ensure that communication with every site is secure. See "Communicating Security Tips."
		The look and feel of web browsing with clientless SSL VPN might be different from what users are accustomed to. For example: • The title bar for clientless SSL VPN appears above each web page. • You access websites by: – Entering the URL in the Enter Web Address field on the clientless SSL VPN Home page – Clicking on a preconfigured website link on the clientless SSL VPN Home page – Clicking a link on a webpage accessed via one of the previous two methods Also, depending on how you configured a particular account, it might be that: • Some websites are blocked • Only the websites that appear as links on the clientless SSL VPN Home page are available
Network Browsing and File Management	File permissions configured for shared remote access	Only shared folders and files are accessible via clientless SSL VPN.
	Server name and passwords for protected file servers	—
	Domain, workgroup, and server names where folders and files reside	Users might not be familiar with how to locate their files through your organization network.
	—	Do not interrupt the **Copy File to Server** command or navigate to a different screen while the copying is in progress. Interrupting the operation can cause an incomplete file to be saved on the server.

Table 39-12 Remote System Configuration and End User Requirements for Clientless SSL VPN (continued)

Task	Remote System or End User Requirements	Specifications or Use Suggestions
Using Application Access	**Note** On Macintosh OS X, only the Safari browser supports this feature.	
	Note Because this feature requires installing Sun Microsystems Java™ Runtime Environment and configuring the local clients, and because doing so requires administrator permissions on the local system, it is unlikely that users will be able to use applications when they connect from public remote systems.	
	Caution Users should always close the Application Access window when they finish using applications by clicking the **Close** icon. Failure to quit the window properly can cause Application Access or the applications themselves to be disabled. See Recovering from hosts File Errors When Using Application Access for details.	
	Client applications installed	—
	Cookies enabled on browser	—
	Administrator privileges	User must have administrator access on the PC if you use DNS names to specify servers because modifying the hosts file requires it.
	Sun Microsystems Java Runtime Environment (JRE) version 1.4.x and 1.5.x installed. Javascript must be enabled on the browser. By default, it is enabled.	If JRE is not installed, a pop-up window displays, directing users to a site where it is available. On rare occasions, the port forwarding applet fails with JAVA exception errors. If this happens, do the following: 1. Clear the browser cache and close the browser. 2. Verify that no JAVA icons are in the computer task bar. Close all instances of JAVA. 3. Establish a clientless SSL VPN session and launch the port forwarding JAVA applet.
	Client applications configured, if necessary. **Note** The Microsoft Outlook client does not require this configuration step. All non-Windows client applications require configuration. To see if configuration is necessary for a Windows application, check the value of the Remote Server. • If the Remote Server contains the server hostname, you do not need to configure the client application. • If the Remote Server field contains an IP address, you must configure the client application.	To configure the client application, use the server's locally mapped IP address and port number. To find this information: 1. Start a clientless SSL VPN session and click the Application Access link on the Home page. The Application Access window appears. 2. In the Name column, find the name of the server you want to use, then identify its corresponding client IP address and port number (in the Local column). 3. Use this IP address and port number to configure the client application. Configuration steps vary for each client application.
	Note Clicking a URL (such as one in an -e-mail message) in an application running over a clientless SSL VPN session does not open the site over that session. To open a site over the session, paste the URL into the Enter Clientless SSL VPN (URL) Address field.	

Table 39-12 Remote System Configuration and End User Requirements for Clientless SSL VPN (continued)

Task	Remote System or End User Requirements	Specifications or Use Suggestions
Using E-mail via Application Access	Fulfill requirements for Application Access (See Using Applications)	To use mail, start Application Access from the clientless SSL VPN Home page. The mail client is then available for use.
	Note If you are using an IMAP client and you lose your mail server connection or are unable to make a new connection, close the IMAP application and restart clientless SSL VPN.	
	Other mail clients	We have tested Microsoft Outlook Express versions 5.5 and 6.0. Clientless SSL VPN should support other SMTPS, POP3S, or IMAP4S e-mail programs via port forwarding, such as Lotus Notes and Eudora, but we have not verified them.
Using E-mail via Web Access	Web-based e-mail product installed	Supported: • Outlook Web Access For best results, use OWA on Internet Explorer 6.x or higher, or Firefox 2.0 or higher. • Lotus iNotes Other web-based e-mail products should also work, but we have not verified them.
Using E-mail via E-mail Proxy (legacy feature)	SSL-enabled mail application installed Do not set the security appliance SSL version to TLSv1 Only. Outlook and Outlook Express do not support TLS.	Supported mail applications: • Microsoft Outlook 2000 and 2002 • Microsoft Outlook Express 5.5 and 6.0 • Eudora 4.2 for Windows 2000 Other SSL-enabled mail clients should also work, but we have not verified them.
	Mail application configured	See instructions and examples for your mail application in "Using E-Mail over Clientless SSL VPN."

Translating the Language of User Messages

The security appliance provides language translation for the portal and screens displayed to users that initiate browser-based, clientless SSL VPN connections, as well as the interface displayed to Cisco AnyConnect VPN Client users.

This section describes how to configure the security appliance to translate these user messages and includes the following sections:

- Understanding Language Translation, page 39-80
- Creating Translation Tables, page 39-81
- Referencing the Language in a Customization Object, page 39-82
- Changing a Group Policy or User Attributes to Use the Customization Object, page 39-84

Understanding Language Translation

Functional areas and their messages that are visible to remote users are organized into translation domains. Table 39-13 shows the translation domains and the functional areas translated.

Table 39-13 Translation Domains and Functional Areas Affected

Translation Domain	Functional Areas Translated
AnyConnect	*Messages displayed on the user interface of the Cisco AnyConnect VPN Client.*
CSD	Messages for Cisco Secure Desktop.
customization	*Messages on the logon and logout pages, portal page, and all the messages customizable by the user.*
banners	Banners displayed to remote users and messages when VPN access is denied.
PortForwarder	Messages displayed to Port Forwarding users.
url-list	Text that user specifies for URL bookmarks on the portal page.
webvpn	All the layer 7, AAA and portal messages that are not customizable.
plugin-ica	Messages for the Citrix plug-in.
plugin-rdp	Messages for the Remote Desktop Protocol plug-in.
plugin-telnet,ssh	Messages for the Telnet and SSH plug-in.
plugin-vnc	Messages for the VNC plug-in.

The software image package for the security appliance includes a translation table template for each domain that is part of the standard functionality. The templates for plug-ins are included with the plug-ins and define their own translation domains.

You can export the template for a translation domain, which creates an XML file of the template at the URL you provide. The message fields in this file are empty. You can edit the messages and import the template to create a new translation table object that resides in flash memory.

You can also export an existing translation table. The XML file created displays the messages you edited previously. Reimporting this XML file with the same language name creates an new version of the translation table object, overwriting previous messages.

Some templates are static, but some change based on the configuration of the security appliance. Because you can customize the *logon and logout pages, portal page, and URL bookmarks for clientless users,* the **security appliance generates the customization** and **url-list** translation domain templates dynamically and the template automatically reflects your changes to these functional areas.

After creating translation tables, they are available to customization objects that you create and apply to group policies or user attributes. With the exception of the AnyConnect translation domain, a translation table has no affect and messages are not translated on user screens until you create a customization object, identify a translation table to use in that object, and specify that customization for the group policy or user. Changes to the translation table for the AnyConnect domain are immediately visible to AnyConnect client users.

Creating Translation Tables

The following procedure describes how to create translation tables:

Step 1 Export a translation table template to a computer with the **export webvpn translation-table** command from privileged EXEC mode.

In the following example, the **show webvpn translation-table** command shows available translation table templates and tables.

```
hostname# show import webvpn translation-table
Translation Tables' Templates:
customization
AnyConnect
CSD
PortForwarder
url-list
webvpn
Citrix-plugin
RPC-plugin
Telnet-SSH-plugin
VNC-plugin

Translation Tables:
```

The next example exports the translation table template for the customization domain, which affects messages displayed for users in clientless SSL VPN sessions. The filename of the XML file created is *portal* (user-specified) and contains empty message fields:

```
hostname# export webvpn translation-table customization template
tftp://209.165.200.225/portal
```

Step 2 Edit the translation table XML file.

The following example shows a portion of the template that was exported as *portal*. The end of this output includes a message ID field (msgid) and a message string field (msgstr) for the message *SSL VPN*, which is displayed on the portal page when a user establishes a clientless SSL VPN session. The complete template contains many pairs of message fields:

```
# Copyright (C) 2006 by Cisco Systems, Inc.
#
#, fuzzy
msgid ""
msgstr ""
"Project-Id-Version: ASA\n"
"Report-Msgid-Bugs-To: vkamyshe@cisco.com\n"
"POT-Creation-Date: 2007-03-12 18:57 GMT\n"
"PO-Revision-Date: YEAR-MO-DA HO:MI+ZONE\n"
"Last-Translator: FULL NAME <EMAIL@ADDRESS>\n"
"Language-Team: LANGUAGE <LL@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=UTF-8\n"
"Content-Transfer-Encoding: 8bit\n"

#: DfltCustomization:24 DfltCustomization:64
msgid "Clientless SSL VPN Service"
msgstr ""
```

The message ID field (msgid) contains the default translation. The message string field (msgstr) that follows msgid provides the translation. To create a translation, enter the translated text between the quotes of the msgstr string.

Step 3 Import the translation table using the **import webvpn translation-table** command from privileged EXEC mode.

In the following example, the XML file is imported *es-us*—the abbreviation for Spanish spoken in the United States.

```
hostname# import webvpn translation-table customization language es-us
tftp://209.165.200.225/portal
hostname# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
hostname# show import webvpn translation-table
Translation Tables' Templates:
AnyConnect
PortForwarder
csd
customization
keepout
url-list
webvpn
Citrix-plugin
RPC-plugin
Telnet-SSH-plugin
VNC-plugin

Translation Tables:
es-us customization
```

If you import a translation table for the AnyConnect domain, your changes are effective immediately. If you import a translation table for any other domain, you must continue to Step 4, where you create a customization object, identify the translation table to use in that object, and specify that customization object for the group policy or user.

Referencing the Language in a Customization Object

Now that you have created a translation table, you need to refer to this table in a customization object.

Steps 4 through 6 describe how to export the customization template, edit it, and import it as a customization object:

Step 4 Export a customization template to a URL where you can edit it using the **export webvpn customization template** command from privileged EXEC mode. The example below exports the template and creates the copy *sales* at the URL specified:

```
hostname# export webvpn customization template tftp://209.165.200.225/sales
```

Step 5 Edit the customization template and reference the previously-imported translation table.

There are two areas of XML code in the customization template that pertain to translation tables. The first area, shown below, specifies the translation tables to use:

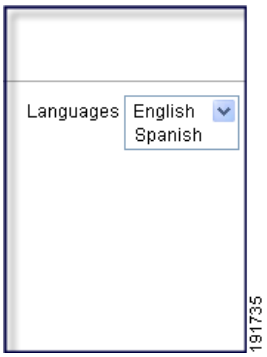
```
<localization>
   <languages>en,ja,zh,ru,ua</languages>
   <default-language>en</default-language>
</localization>
```

The <languages> tag in the XML code is followed by the names of the translation tables. In this example, they are en, ja, zh, ru, and ua. For the customization object to call these translation tables correctly, the tables must have been previously imported using the same names. These names must be compatible with language options of the browser.

The <default-language> tag specifies the language that the remote user first encounters when connecting to the security appliance. In the example code above, the language is English.

Figure 39-16 shows the Language Selector that displays on the logon page. The Language Selector gives the remote user establishing an SSL VPN connection the ability to choose a language.

Figure 39-16 Language Selector



The XML code below affects the display of the Language Selector, and includes the <language selector> tag and the associated <language> tags that enable and customize the Language Selector:

```
<auth-page>
     ....
         <language-selector>
            <mode>enable</mode>
            <title l10n="yes">Language:</title>
          <language>
             <code>en</code>
             <text>English</text>
          </language>
          <language>
            <code>es-us</code>
            <text>Spanish</text>
          </language>
       </language-selector>
```

The <language-selector> group of tags includes the <mode> tag that enables and disables the displaying of the Language Selector, and the <title> tag that specifies the title of the drop-down box listing the languages.

The <language> group of tags includes the <code> and <text> tags that map the language name displayed in the Language Selector drop-down box to a specific translation table.

Make your changes to this file and save the file.

Step 6 Import the customization template as a new object using the **import webvpn customization** command from privileged EXEC mode. For example:

```
hostname# import webvpn customization sales tftp://209.165.200.225/sales
hostname# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

The output of the **show import webvpn customization** command shows the new customization object *sales*:

```
hostname(config)# show import webvpn customization
Template
sales
hostname(config)#
```

Changing a Group Policy or User Attributes to Use the Customization Object

Now that you have created the customization object, you need to activate your changes for specific groups or users. Step 7 shows how to enable the customization object in a group policy:

Step 7 Enter the group policy webvpn configuration mode for a group policy and enable the customization object using the **customization** command. The following example shows the customization object *sales* enabled in the group policy *sales*:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# customization value sales
```

Capturing Data

The CLI **capture** command lets you log information about websites that do not display properly over a clientless SSL VPN session. This data can help your Cisco customer support engineer troubleshoot problems. The following sections describe how to capture and view clientless SSL VPN session data:

- Creating a Capture File
- Using a Browser to Display Capture Data

Note Enabling clientless SSL VPN capture affects the performance of the security appliance. Be sure to disable the capture after you generate the capture files needed for troubleshooting.

Creating a Capture File

Perform the following steps to capture data about a clientless SSL VPN session to a file.

Step 1 To start the capture utility for clientless SSL VPN, use the **capture** command from privileged EXEC mode.

capture *capture_name* **type webvpn user** *webvpn_username*

where:

- *capture_name* is a name you assign to the capture, which is also prepended to the name of the capture files.
- *webvpn_user* is the username to match for capture.

The capture utility starts.

Step 2 A user logs in to begin a clientless SSL VPN session. The capture utility is capturing packets.

Stop the capture by using the **no** version of the command.

no capture *capture_name*

Note You must use the **no capture** command before webvpn logoff.

The capture utility creates a *capture_name*.zip file, which is encrypted with the password **koleso**.

Step 3 Send the .zip file to Cisco Systems, or attach it to a Cisco TAC service request.

Step 4 To look at the contents of the .zip file, unzip it using the password **koleso**.

The following example creates a capture named *hr*, which captures traffic for user2 to a file:

```
hostname# capture hr type webvpn user user2
WebVPN capture started.
   capture name   hr
   user name      user2
hostname# no capture hr
```

Using a Browser to Display Capture Data

Perform the following steps to capture data about a clientless SSL VPN session and view it in a browser.

Step 1 To start the capture utility for clientless SSL VPN, use the **capture** command from privileged EXEC mode.

capture *capture_name* **type webvpn user** *webvpn_username*

where:

- *capture_name* is a name you assign to the capture, which is also prepended to the name of the capture files.
- *webvpn_user* is the username to match for capture.

The capture utility starts.

Step 2 A user logs in to begin a clientless SSL VPN session. The capture utility is capturing packets.

Stop the capture by using the **no** version of the command.

Step 3 Open a browser and in the address box enter

https://*asdm_enabled_interface_of_the_security_appliance***:***port***/admin/capture/***capture_name*/pcap

The following example command displays the capture named hr:

https://192.0.2.1:60000/admin/capture/hr/pcap

The captured content displays in a sniffer format.

Step 4 When you finish examining the capture content, stop the capture by using the **no** version of the command.



CHAPTER 40

Configuring AnyConnect VPN Client Connections

The Cisco AnyConnect SSL VPN Client provides secure SSL connections to the security appliance for remote users. Without a previously-installed client, remote users enter the IP address in their browser of an interface configured to accept SSL VPN connections. Unless the security appliance is configured to redirect http:// requests to https://, users must enter the URL in the form https://<*address*>.

After entering the URL, the browser connects to that interface and displays the login screen. If the user satisfies the login and authentication, and the security appliance identifies the user as requiring the client, it downloads the client that matches the operating system of the remote computer. After downloading, the client installs and configures itself, establishes a secure SSL connection and either remains or uninstalls itself (depending on the security appliance configuration) when the connection terminates.

In the case of a previously installed client, when the user authenticates, the security appliance examines the revision of the client, and upgrades the client as necessary.

When the client negotiates an SSL VPN connection with the security appliance, it connects using Transport Layer Security (TLS), and optionally, Datagram Transport Layer Security (DTLS). DTLS avoids latency and bandwidth problems associated with some SSL connections and improves the performance of real-time applications that are sensitive to packet delays.

The AnyConnect client can be downloaded from the security appliance, or it can be installed manually on the remote PC by the system administrator. For more information about installing the client manually, see the *Cisco AnyConnect VPN Client Administrator Guide*.

The security appliance downloads the client based on the group policy or username attributes of the user establishing the connection. You can configure the security appliance to automatically download the client, or you can configure it to prompt the remote user about whether to download the client. In the latter case, if the user does not respond, you can configure the security appliance to either download the client after a timeout period or present the login page.

This section covers the following topics:

- Installing the AnyConnect SSL VPN Client, page 40-2
- Enabling AnyConnect Client Connections, page 40-3
- Enabling Permanent Client Installation, page 40-5
- Configuring DTLS, page 40-5
- Prompting Remote Users, page 40-6
- Enabling AnyConnect Client Profile Downloads, page 40-6
- Enabling Additional AnyConnect Client Features, page 40-8
- Configuring Advanced SSL VPN Features, page 40-12

Installing the AnyConnect SSL VPN Client

This section presents the platform requirements and the procedure for installing the AnyConnect client on the security appliance and preparing it for download to remote users.

Remote PC System Requirements

The AnyConnect client supports the following operating systems on the remote PC:

- Microsoft Visa
- Microsoft Windows 2000
- Microsoft Windows XP
- MAC Intel
- MAC Power PC
- Linux

The legacy SSL VPN Client (SVC) supports the following operating systems on the remote PC:

- Microsoft Windows 2000
- Microsoft Windows XP

Installing the AnyConnect Client

Installing the client on the security appliance consists of copying a client image to the security appliance and identifying the file as a client image. With multiple clients, you must also assign the order that the security appliance downloads the clients to the remote PC. Perform the following steps to install the client:

Step 1 Copy the client image package to the security appliance using the **copy** command from privileged EXEC mode, or using another method. In this example, the images are copied from a tftp server using the **copy tftp** command:

```
hostname# copy tftp flash
Address or name of remote host []? 209.165.200.226
Source filename []? anyconnect-win-2.2.0128-k9.pkg
Destination filename []? sslclient-win-2.2.0128.pkg
Accessingtftp://209.165.200.226/anyconnect-win-2.2.0128-k9.pkg...!!!!!!!!!!!!!!!!!!!!!!!
Writing file
disk0:/cdisk71...!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
319662 bytes copied in 3.695 secs (86511 bytes/sec)
```

Step 2 Identify a file on flash as an SSL VPN client package file using the **svc image** command from webvpn configuration mode:

svc image *filename order*

The security appliance expands the file in cache memory for downloading to remote PCs. If you have multiple clients, assign an order to the client images with the order argument.

The security appliance downloads portions of each client in the order you specify until it matches the operating system of the remote PC. Therefore, assign the lowest number to the image used by the most commonly-encountered operating system. For example:

```
hostname(config-webvpn)# svc image anyconnect-win-2.2.0128-k9.pkg 1
hostname(config-webvpn)# svc image anyconnect-macosx-i386-2.2.0128-k9.pkg 2
hostname(config-webvpn)# svc image anyconnect-linux-2.2.0128-k9.pkg 3
```

Note The security appliance expands SSL VPN client and the CSD images in cache memory. If you receive the error message *ERROR: Unable to load SVC image - increase disk space via the 'cache-fs' command*, use the **cache-fs limit** command to adjust the size of cache memory:

Step 3 Check the status of the clients using the **show webvpn svc** command:

```
hostname(config-webvpn)# show webvpn svc
1. disk0:/anyconnect-win-2.2.0128-k9.pkg 1
  CISCO STC win2k+
  2,0,0310
  Tue 03/27/2007  4:16:21.09

2. disk0:/anyconnect-macosx-i386-2.2.0128-k9.pkg 2
  CISCO STC Darwin_i386
  2,0,0
  Tue Mar 27 05:09:16 MDT 2007

3. disk0:/anyconnect-linux-2.2.0128-k9.pkg 3
  CISCO STC Linux
  2,0,0
  Tue Mar 27 04:06:53 MST 2007

3 SSL VPN Client(s) installed
```

Enabling AnyConnect Client Connections

After installing the client, enable the security appliance to allow SSL VPN client connections by performing the following steps:

Step 1 Enable clientless connections on an interface using the **enable** command from webvpn mode:

enable *interface*

For example:

```
hostname(config)# webvpn
hostname(config-webvpn)# enable outside
```

Step 2 Configure a method of address assignment. You can use DHCP, and/or user-assigned addressing. You can also create a local IP address pool using the **ip local pool** command from global configuration mode:

ip local pool *poolname startaddr-endaddr* **mask** *mask*

The following example creates the local IP address pool *vpn_users*:

```
hostname(config)# ip local pool vpn_users 209.165.200.225-209.165.200.254
mask 255.255.255.224
```

Step 3 Assign IP addresses to a tunnel group. One method you can use to do this is to assign a local IP address pool with the **address-pool** command from general-attributes mode:

address-pool *poolname*

To do this, first enter the **tunnel-group** *name* **general-attributes** command to enter general-attributes mode. Then specify the local IP address pool using the **address-pool** command.

In the following example, the user configures the existing tunnel group *telecommuters* to use the address pool *vpn_users created in step 3:*

```
hostname(config)# tunnel-group telecommuters general-attributes
hostname(config-tunnel-general)# address-pool vpn_users
```

Step 4 Assign a default group policy to the tunnel group with the **default-group-policy** command from tunnel group general attributes mode:

default-group-policy *name*

In the following example, the user assigns the group policy *sales* to the tunnel group *telecommuters*:

```
hostname(config-tunnel-general)# default-group-policy sales
```

Step 5 Create and enable a group alias that displays in the group list on the WebVPN Login page using the **group-alias** command from tunnel group webvpn attributes mode:

group-alias *name* **enable**

First exit to global configuration mode, and then enter the **tunnel-group** *name* **webvpn-attributes** command to enter tunnel group webvpn attributes mode.

In the following example, the user enters webvpn attributes configuration mode for the tunnel group *telecommuters*, and creates the group alias *sales_department*:

```
hostname(config)# tunnel-group telecommuters webvpn-attributes
hostname(config-tunnel-webvpn)# group-alias sales_department enable
```

Step 6 Enable the display of the tunnel-group list on the WebVPN Login page from webvpn mode:

tunnel-group-list enable

First exit to global configuration mode, and then enter webvpn mode.

In the following example, the user enters webvpn mode, and then enables the tunnel group list:

```
hostname(config)# webvpn
hostname(config-webvpn)# tunnel-group-list enable
```

Step 7 Specify SSL as a permitted VPN tunneling protocol for the group or user with the **vpn-tunnel-protocol svc** command in group-policy mode or username mode. You can also specify additional protocols. For more information, see the vpn-tunnel-protocol command in the *Cisco ASA 5500 Series Command Reference*.

vpn-tunnel-protocol svc

To do this, first exit to global configuration mode, enter the **group-policy** *name* **attributes** command to enter group-policy mode, or the **username** *name* **attributes** command to enter username mode, and then enter the **webvpn** command to enter webvpn mode and change the WebVPN settings for the group or user.

The following example identifies SSL as the only permitted tunneling protocol for the group-policy *sales*:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# vpn-tunnel-protocol svc
```

For more information about assigning users to group policies, see Chapter 32, “Configuring Connection Profiles, Group Policies, and Users”.

Enabling Permanent Client Installation

Enabling permanent client installation disables the automatic uninstalling feature of the client. The client remains installed on the remote computer for subsequent connections, reducing the connection time for the remote user.

To enable permanent client installation for a specific group or user, use the **svc keep-installer** command from group-policy or username webvpn modes:

svc keep-installer installed

The default is that permanent installation of the client is disabled. The client on the remote computer uninstalls at the end of every session. The following example configures the existing group-policy *sales* to keep the client installed on the remote computer:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-policy)# svc keep-installer installed
```

Configuring DTLS

Datagram Transport Layer Security (DTLS) allows the AnyConnect client establishing an SSL VPN connection to use two simultaneous tunnels—an SSL tunnel and a DTLS tunnel. Using DTLS avoids latency and bandwidth problems associated with SSL connections and improves the performance of real-time applications that are sensitive to packet delays.

By default, DTLS is enabled when SSL VPN access is enabled on an interface. If you disable DTLS, SSL VPN connections connect with an SSL VPN tunnel only.

Note In order for DTLS to fall back to a TLS connection, Dead Peer Detection (DPD) must be enabled. If you do not enable DPD, and the DTLS connection experiences a problem, the connection terminates instead of falling back to TLS. For more information on enabling DPD, see Enabling and Adjusting Dead Peer Detection, page 40-12

You can disable DTLS for all AnyConnect client users with the **enable** command **tls-only** option in webvpn configuration mode:

enable <*interface*> **tls-only**

For example:

hostname(config-webvpn)# **enable outside tls-only**

By default, DTLS is enabled for specific groups or users with the **svc dtls enable** command in group policy webvpn or username webvpn configuration mode:

[**no**] **svc dtls enable**

If you need to disable DTLS, use the **no** form of the command. For example:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# no svc dtls enable
```

Prompting Remote Users

You can enable the security appliance to prompt remote SSL VPN client users to download the client with the **svc ask** command from group policy webvpn or username webvpn configuration modes:

[**no**] **svc ask** {**none** | **enable** [**default** {**webvpn** | **svc**} **timeout** *value*]}

svc ask enable prompts the remote user to download the client or go to the clientless portal page and waits indefinitely for user response.

svc ask enable default svc immediately downloads the client.

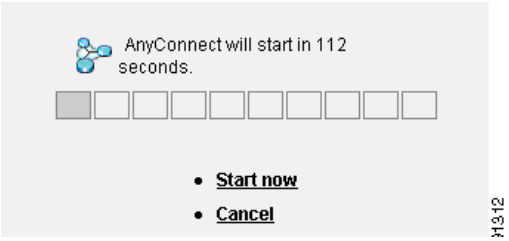
svc ask enable default webvpn immediately goes to the portal page.

svc ask enable default svc timeout *value* prompts the remote user to download the client or go to the clientless portal page and waits the duration of *value* before taking the default action—downloading the client.

svc ask enable default clientless timeout *value* prompts the remote user to download the client or go to the clientless portal page, and waits the duration of *value* before taking the default action—displaying the clientless portal page.

Figure 40-1 shows the prompt displayed to remote users when either **default svc timeout** *value or* **default webvpn timeout** *value is configured:*

Figure 40-1 Prompt Displayed to Remote Users for SSL VPN Client Download



The following example configures the security appliance to prompt the user to download the client or go to the clientless portal page and wait *10 seconds for a response* before downloading the client:

```
hostname(config-group-webvpn)# svc ask enable default svc timeout 10
```

Enabling AnyConnect Client Profile Downloads

An AnyConnect client profile is a group of configuration parameters, stored in an XML file, that the client uses to configure the connection entries that appear in the client user interface. These parameters (XML tags) include the names and addresses of host computers and settings to enable additional client features.

The AnyConnect client installation includes a profile template, named *AnyConnectProfile.tmpl*, that you can edit with a text editor and use as a basis to create other profile files. You can also set advanced parameters that are not available through the user interface. The installation also includes a complete XML schema file, named *AnyConnectProfile.xsd*.

After creating a profile, you must load the file on the security appliance and configure the security appliance to download it to remote client PCs.

Follow these steps to edit a profile and enable the security appliance to download it to remote clients:

Step 1 Retrieve a copy of the profile file (AnyConnectProfile.tmpl) from a client installation. Table 40-1 shows the installation path for each operating system.

Table 40-1 Operating System and Profile File Installation Path

Operating System	Installation Path
Windows Vista	%ALLUSERSPROFILE%\Cisco\Cisco AnyConnect VPN Client\Profile[1]
Windows XP and 2000	%ALLUSERSPROFILE%/Application Data/Cisco/Cisco AnyConnect VPN Client/Profile[2]
Linux	/opt/cisco/vpn/profile
Mac OS X	/opt/cisco/vpn/profile

1. %ALLUSERSPROFILE% refers to the environmental variable by the same name for Windows Vista. In most installations, this is C:\Program Files.
2. %PROGRAMFILES% refers to the environmental variable by the same name for Windows XP and 2000. In most installations, this is C:\Program Files.

Step 2 Edit the profile file. The example below shows the contents of the profile file (AnyConnectProfile.tmpl) for Windows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    This is a template file that can be configured to support the
    identification of secure hosts in your network.

    The file needs to be renamed to cvcprofile.xml (for now).

    There is an ASA command to import updated profiles for downloading to
    client machines.  Provide some basic instruction.....
-->
<Configuration>
    <ClientInitialization>
        <UseStartBeforeLogon>false</UseStartBeforeLogon>
    </ClientInitialization>
    <HostProfile>
        <HostName></HostName>
        <HostAddress></HostAddress>
    </HostProfile>
    <HostProfile>
        <HostName></HostName>
        <HostAddress></HostAddress>
    </HostProfile>
</Configuration>
```

The <HostProfile> tags are frequently edited so that the AnyConnect client displays the names and addresses of host computers for remote users. The following example shows the <HostName> and <HostAddress> tags, with the name and address of a host computer inserted:

```
    <HostProfile>
        <HostName>Sales_gateway</HostName>
        <HostAddress>209.165.200.225</HostAddress>
    </HostProfile>
```

Step 3 Load the profile file into flash memory on the security appliance and then use the **svc profiles** command from webvpn configuration mode to identify the file as a client profile to load into cache memory:

[no] svc profiles *name path}*

After the file is loaded into cache memory, the profile is available to group policies and username attributes of client users.

In the following example, the user previously created two new profile files (sales_hosts.xml and engineering_hosts.xml) from the AnyConnectProfile.tmpl file provided in the client installation and uploaded them to flash memory. Then the user specifies these files as profiles for use by group policies, specifying the names *sales* and *engineering*:

```
asa1(config-webvpn)# svc profiles sales disk0:/sales_hosts.xml
asa1(config-webvpn)# svc profiles engineering disk0:/engineering_hosts.xml
```

Entering the **dir cache:stc/profiles** command shows the profiles loaded into cache memory:

```
hostname(config-webvpn)# dir cache:/stc/profiles

Directory of cache:stc/profiles/

0      ----  774         11:54:41 Nov 22 2006  engineering.xml
0      ----  774         11:54:29 Nov 22 2006  sales.xml

2428928 bytes total (18219008 bytes free)
hostname(config-webvpn)#
```

Step 4 Enter group policy webvpn or username attributes webvpn configuration mode and specify a profile for the group or user with the **svc profiles** command:

[no] svc profiles *{***value** *profile* | **none***}*

In the following example, the user follows the **svc profiles value** command with a question mark (**?**) view the available profiles. Then the user configures the group policy to use the profile *sales*:

```
asa1(config-group-webvpn)# svc profiles value ?

config-group-webvpn mode commands/options:
Available configured profile packages:
  engineering
  sales
asa1(config-group-webvpn)# svc profiles sales
asa1(config-group-webvpn)#
```

Enabling Additional AnyConnect Client Features

To minimize download time, the client only requests downloads (from the security appliance) of the core modules that it needs. As additional features become available for the AnyConnect client, you need to update the remote clients in order for them to use the features.

To enable new features, you must specify the new module names using the **svc modules** command from group policy webvpn or username webvpn configuration mode:

[no] svc modules {**none** | **value** *string*}

Separate multiple strings with commas.

For a list of values to enter for each client feature, see the release notes for the Cisco AnyConnect VPN Client.

Enabling Start Before Logon

Start Before Logon (SBL) allows login scripts, password caching, drive mapping, and more, for the AnyConnect client installed on a Windows PC. For SBL, you must enable the security appliance to download the module which enables graphical identification and authentication (GINA) for the AnyConnect client. The following procedure shows how to enable SBL:

Step 1 Enable the security appliance to download the GINA module for VPN connection to specific groups or users using the **svc modules** *vpngina* command from group policy webvpn or username webvpn configuration modes.

In the following example, the user enters group-policy attributes mode for the group policy *telecommuters*, enters webvpn configuration mode for the group policy, and specifies the string *vpngina*:

```
hostname(config)# group-policy telecommuters attributes
hostname(config-group-policy)# webvpn
hostame(config-group-webvpn)# svc modules value vpngina
```

Step 2 Retrieve a copy of the client profiles file (AnyConnectProfile.tmpl). For information on the location of the profiles file for each operating system, see Table 40-1 on page 40-7

Step 3 Edit the profiles file to specify that SBL is enabled. The example below shows the relevant portion of the profiles file (AnyConnectProfile.tmpl) for Windows:

```
<Configuration>
    <ClientInitialization>
        <UseStartBeforeLogon>false</UseStartBeforeLogon>
    </ClientInitialization>
```

The <UseStartBeforeLogon> tag determines whether the client uses SBL. To turn SBL on, replace *false* with *true*. The example below shows the tag with SBL turned on:

```
    <ClientInitialization>
        <UseStartBeforeLogon>true</UseStartBeforeLogon>
    </ClientInitialization>
```

Step 4 Save the changes to AnyConnectProfile.tmpl and update the profile file for the group or user on the security appliance using the **svc profile** command from webvpn configuration mode. For example:

```
asa1(config-webvpn)# svc profiles sales disk0:/sales_hosts.xml
```

Translating Languages for AnyConnect User Messages

The security appliance provides language translation for the portal and screens displayed to users that initiate browser-based, Clientless SSL VPN connections, as well as the interface displayed to Cisco AnyConnect VPN Client users.

This section describes how to configure the security appliance to translate these user messages and includes the following sections:

- Understanding Language Translation, page 40-10
- Creating Translation Tables, page 40-10

Understanding Language Translation

Functional areas and their messages that are visible to remote users are organized into translation domains. All *messages displayed on the user interface of the Cisco AnyConnect VPN Client are located in the AnyConnect domain.*

The software image package for the security appliance includes a translation table template for the AnyConnect domain. You can export the template, which creates an XML file of the template at the URL you provide. The message fields in this file are empty. You can edit the messages and import the template to create a new translation table object that resides in flash memory.

You can also export an existing translation table. The XML file created displays the messages you edited previously. Reimporting this XML file with the same language name creates an new version of the translation table object, overwriting previous messages. Changes to the translation table for the AnyConnect domain are immediately visible to AnyConnect client users.

Creating Translation Tables

The following procedure describes how to create translation tables for the AnyConnect domain:

Step 1 Export a translation table template to a computer with the **export webvpn translation-table** command from privileged EXEC mode.

In the following example, the **show webvpn translation-table** command shows available translation table templates and tables.

```
hostname# show import webvpn translation-table
Translation Tables' Templates:
customization
AnyConnect
CSD
PortForwarder
url-list
webvpn
Citrix-plugin
RPC-plugin
Telnet-SSH-plugin
VNC-plugin

Translation Tables:
```

Then the user exports the translation table for the AnyConnect translation domain. The filename of the XML file created is named *client* and contains empty message fields:

```
hostname# export webvpn translation-table AnyConnect template
tftp://209.165.200.225/client
```

In the next example, the user exports a translation table named *zh*, which was previously imported from a template. zh is the abbreviation by Microsoft Internet Explorer for the Chinese language.

```
hostname# export webvpn translation-table customization language zh
tftp://209.165.200.225/chinese_client
```

Step 2 Edit the Translation Table XML file. The following example shows a portion of the AnyConnect template. The end of this output includes a message ID field (msgid) and a message string field (msgstr) for the message *Connected*, which is displayed on the AnyConnect client GUI when the client establishes a VPN connection. The complete template contains many pairs of message fields:

```
# SOME DESCRIPTIVE TITLE.
```

```
# Copyright (C) YEAR THE PACKAGE'S COPYRIGHT HOLDER
# This file is distributed under the same license as the PACKAGE package.
# FIRST AUTHOR <EMAIL@ADDRESS>, YEAR.
#
#, fuzzy
msgid ""
msgstr ""
"Project-Id-Version: PACKAGE VERSION\n"
"Report-Msgid-Bugs-To: \n"
"POT-Creation-Date: 2006-11-01 16:39-0700\n"
"PO-Revision-Date: YEAR-MO-DA HO:MI+ZONE\n"
"Last-Translator: FULL NAME <EMAIL@ADDRESS>\n"
"Language-Team: LANGUAGE <LL@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=CHARSET\n"
"Content-Transfer-Encoding: 8bit\n"

#: C:\cygwin\home\cafitz\cvc\main\Api\AgentIfc.cpp:23
#: C:\cygwin\home\cafitz\cvc\main\Api\check\AgentIfc.cpp:22
#: C:\cygwin\home\cafitz\cvc\main\Api\save\AgentIfc.cpp:23
#: C:\cygwin\home\cafitz\cvc\main\Api\save\AgentIfc.cpp~:20
#: C:\cygwin\home\cafitz\cvc\main\Api\save\older\AgentIfc.cpp:22
msgid "Connected"
msgstr ""
```

The msgid contains the default translation. The msgstr that follows msgid provides the translation. To create a translation, enter the translated text between the quotes of the msgstr string. For example, to translate the message “Connected” with a Spanish translation, insert the Spanish text between the quotes:

```
msgid "Connected"
msgstr "Conectado"
```

Be sure to save the file.

Step 3 Import the translation table using the **import webvpn translation-table** command from privileged EXEC mode. Be sure to specify the name of the new translation table with the abbreviation for the language that is compatible with the browser.

In the following example, the XML file is imported *es-us*—the abbreviation used by Microsoft Internet Explorer for Spanish spoken in the United States.

```
hostname# import webvpn translation-table AnyConnect language es-us
tftp://209.165.200.225/client
hostname# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
hostname# show import webvpn translation-table
Translation Tables' Templates:
AnyConnect
PortForwarder
csd
customization
keepout
url-list
webvpn
Citrix-plugin
RPC-plugin
Telnet-SSH-plugin
VNC-plugin

Translation Tables:
es-us AnyConnect
```

Configuring Advanced SSL VPN Features

The following section describes advanced features that fine-tune SSL VPN connections, and includes the following sections:

- Enabling Rekey, page 40-12
- Enabling and Adjusting Dead Peer Detection, page 40-12
- Enabling Keepalive, page 40-13
- Using Compression, page 40-14
- Adjusting MTU Size, page 40-14
- Viewing SSL VPN Sessions, page 40-15
- Logging Off SVC Sessions, page 40-15
- Updating SSL VPN Client Images, page 40-16

Enabling Rekey

When the security appliance and the SSL VPN client perform a rekey, they renegotiate the crypto keys and initialization vectors, increasing the security of the connection.

To enable the client to perform a rekey on an SSL VPN connection for a specific group or user, use the **svc rekey** command from group-policy and username webvpn modes.

[no] svc rekey {**method** {**new-tunnel** | **none** | **ssl**} | **time** *minutes*}

method new-tunnel specifies that the client establishes a new tunnel during rekey.

method none disables rekey.

method ssl specifies that SSL renegotiation takes place during rekey.

time *minutes* specifies the number of minutes from the start of the session, or from the last rekey, until the rekey takes place, from 1 to 10080 (1 week).

In the following example, the client is configured to renegotiate with SSL during rekey, which takes place 30 minutes after the session begins, for the existing group-policy *sales*:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-policy)# svc rekey method ssl
hostname(config-group-policy)# svc rekey time 30
```

Enabling and Adjusting Dead Peer Detection

Dead Peer Detection (DPD) ensures that the security appliance (gateway) or the client can quickly detect a condition where the peer is not responding, and the connection has failed.

To enable DPD on the security appliance or client for a specific group or user, and to set the frequency with which either the security appliance or client performs DPD, use the **svc dpd-interval** command from group-policy or username webvpn mode:

svc dpd-interval {[**gateway** {*seconds* | **none***}*] | [**client** {*seconds* | **none***}*]}

no svc dpd-interval {[**gateway** {*seconds* | **none***}*] | [**client** {*seconds* | **none***}*]}

Where:

gateway seconds enables DPD performed by the security appliance (gateway) and specifies the frequency, from 5 to 3600 seconds, with which the security appliance (gateway) performs DPD.

gateway none disables DPD performed by the security appliance.

client *seconds* enable DPD performed by the client, and specifies the frequency, from 5 to 3600 seconds, with which the client performs DPD.

client none disables DPD performed by the client.

To remove the **svc dpd-interval** command from the configuration, use the **no** form of the command:

Note If you enable DTLS, enable Dead Peer Detection (DPD) also. DPD enables a failed DTLS connection to fallback to TLS. Overwise, the connection terminates.

The following example sets the frequency of DPD performed by the security appliance to 30 seconds, and the frequency of DPD performed by the client set to 10 seconds for the existing group-policy *sales*:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-policy)# svc dpd-interval gateway 30
hostname(config-group-policy)# svc dpd-interval client 10
```

Enabling Keepalive

You can adjust the frequency of keepalive messages to ensure that an SSL VPN connection through a proxy, firewall, or NAT device remains open, even if the device limits the time that the connection can be idle. Adjusting the frequency also ensures that the client does not disconnect and reconnect when the remote user is not actively running a socket-based application, such as Microsoft Outlook or Microsoft Internet Explorer.

To set the frequency of keepalive messages, use the **svc keepalive** command from group-policy webvpn or username webvpn configuration mode:

[no] svc keepalive {none | *seconds***}**

none disables client keepalive messages.

seconds enables the client to send keepalive messages, and specifies the frequency of the messages in the range of 15 to 600 seconds.

The default is keepalive messages are disabled.

Use the **no** form of the command to remove the command from the configuration and cause the value to be inherited:

In the following example, the security appliance is configured to enable the client to send keepalive messages with a frequency of 300 seconds (5 minutes), for the existing group-policy *sales*:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# svc keepalive 300
```

Using Compression

Compression increases the communications performance between the security appliance and the client by reducing the size of the packets being transferred for low-bandwidth connections. By default, compression for all SSL VPN connections is enabled on the security appliance, both at the global level and for specific groups or users.

Compression must be turned-on globally using the **compression svc** command from global configuration mode, and then it can be set for specific groups or users with the **svc compression** command in group-policy and username webvpn modes.

Changing Compression Globally

To change the global compression settings, use the **compression svc** command from global configuration mode:

compression svc

no compression svc

To remove the command from the configuration, use the **no** form of the command.

In the following example, compression is disabled for all SSL VPN connections globally:

```
hostname(config)# no compression svc
```

Changing Compression for Groups and Users

To change compression for a specific group or user, use the **svc compression** command in the group-policy and username webvpn modes:

svc compression {**deflate** | **none**}

no svc compression {**deflate** | **none**}

By default, for groups and users, SSL compression is set to *deflate* (enabled).

To remove the **svc compression** command from the configuration and cause the value to be inherited from the global setting, use the **no** form of the command:

In the following example, compression is disabled for the group-policy sales:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# svc compression none
```

Adjusting MTU Size

You can adjust the MTU size (from 256 to 1406 bytes) for SSL VPN connections established by the client with the **svc mtu** command from group policy webvpn or username webvpn configuration mode:

[no] svc mtu *size*

This command affects only the AnyConnect client. The legacy Cisco SSL VPN Client (SVC) is not capable of adjusting to different MTU sizes.

The default for this command in the default group policy is **no svc mtu**. The MTU size is adjusted automatically based on the MTU of the interface that the connection uses, minus the IP/UDP/DTLS overhead.

This command affects client connections established in SSL and those established in SSL with DTLS.

Examples The following example configures the MTU size to 1200 bytes for the group policy *telecommuters*:

```
hostname(config)# group-policy telecommuters attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# svc mtu 1200
```

Viewing SSL VPN Sessions

You can view information about active sessions using the **show vpn-sessiondb** command in privileged EXEC mode:

show vpn-sessiondb svc

The Inactivity field shows the elapsed time since an AnyConnect session lost connectivity. If the session is active, 00:00m:00s appears in this field.

The following example shows the output of the **show vpn-sessiondb svc** command:

```
hostname# show vpn-sessiondb svc

Session Type: SSL VPN Client

Username     : lee
Index        : 1                      IP Addr      : 209.165.200.232
Protocol     : SSL VPN Client         Encryption   : 3DES
Hashing      : SHA1                   Auth Mode    : userPassword
TCP Dst Port : 443                    TCP Src Port : 54230
Bytes Tx     : 20178                  Bytes Rx     : 8662
Pkts Tx      : 27                     Pkts Rx      : 19
Client Ver   : Cisco STC 1.1.0.117
Client Type  : Internet Explorer
Group        : DfltGrpPolicy
Login Time   : 14:32:03 UTC Wed Mar 20 2007
Duration     : 0h:00m:04s
Inactivity   : 0h:28m:48s
Filter Name  :
```

Logging Off SVC Sessions

To log off all SSL VPN sessions, use the **vpn-sessiondb logoff svc** command in global configuration mode:

vpn-sessiondb logoff svc

The following example logs off all SSL VPN sessions:

```
hostname# vpn-sessiondb logoff svc
INFO: Number of sessions of type "svc" logged off : 1
```

You can log off individual sessions using either the **name option, or the index option**:

vpn-session-db logoff name *name*

vpn-session-db logoff index *index*

You can find both the username and the index number (established by the order of the client images) in the output of the **show vpn-sessiondb svc** command. The following example shows the username *lee* and index number *1*.

The Inactivity field shows the elapsed time since an AnyConnect session lost connectivity. If the session is active, 00:00m:00s appears in this field.

```
hostname# show vpn-sessiondb svc

Session Type: SSL VPN Client

Username     : lee
Index        : 1                      IP Addr      : 209.165.200.232
Protocol     : SSL VPN Client         Encryption   : 3DES
Hashing      : SHA1                   Auth Mode    : userPassword
TCP Dst Port : 443                    TCP Src Port : 54230
Bytes Tx     : 20178                  Bytes Rx     : 8662
Pkts Tx      : 27                     Pkts Rx      : 19
Client Ver   : Cisco STC 1.1.0.117
Client Type  : Internet Explorer
Group        : DfltGrpPolicy
Login Time   : 14:32:03 UTC Wed Mar 26 2007
Duration     : 0h:28m:48s
Inactivity   : 0h:00m:04s
Filter Name  :
```

The following example terminates the session using the **name** option of the **vpn-session-db logoff command**:

```
hostname# vpn-sessiondb logoff name tester
Do you want to logoff the VPN session(s)? [confirm]
INFO: Number of sessions with name "mkrupp" logged off : 0

hostname#
```

When the ASA receives the new connection, the session that has been inactive the longest is logged off, allowing new connections to take its license.

Updating SSL VPN Client Images

You can update the client images on the security appliance at any time using the following procedure:

Step 1 Copy the new client images to the security appliance using the **copy** command from privileged EXEC mode, or using another method.

Step 2 If the new client image files have the same filenames as the files already loaded, reenter the **svc image** command that is in the configuration. If the new filenames are different, uninstall the old files using the **no svc image** command. Then use the **svc image** command to assign an order to the images and cause the security appliance to load the new images.



CHAPTER 41

Configuring Certificates

This chapter describes how to configure certificates. CAs are responsible for managing certificate requests and issuing digital certificates. A digital certificate contains information that identifies a user or device. Some of this information can include a name, serial number, company, department, or IP address. A digital certificate also contains a copy of the public key for the user or device. A CA can be a trusted third party, such as VeriSign, or a private (in-house) CA that you establish within your organization.

This chapter includes the following sections:

- Public Key Cryptography, page 41-1
- Certificate Configuration, page 41-5
- The Local CA, page 41-17

Public Key Cryptography

This section includes the following topics:

- About Public Key Cryptography, page 41-1
- Certificate Scalability, page 41-2
- About Key Pairs, page 41-2
- About Trustpoints, page 41-3
- About CRLs, page 41-3
- Supported CA Servers, page 41-5

About Public Key Cryptography

Digital signatures, enabled by public key cryptography, provide a means to authenticate devices and users. In public key cryptography, such as the RSA encryption system, each user has a key pair containing both a public and a private key. The keys act as complements, and anything encrypted with one of the keys can be decrypted with the other.

In simple terms, a signature is formed when data is encrypted with a private key. The signature is attached to the data and sent to the receiver. The receiver applies the public key of the sender to the data. If the signature sent with the data matches the result of applying the public key to the data, the validity of the message is established.

This process relies on the receiver having a copy of the public key of the sender and having a high degree of certainty that this key belongs to the sender, not to someone pretending to be the sender.

Obtaining the public key of a sender is normally handled out-of-band or through an operation done at installation. For instance, most web browsers are configured with the root certificates of several CAs by default. For VPN, the IKE protocol, a component of IPSec, can use digital signatures to authenticate peer devices before setting up security associations.

Certificate Scalability

Without digital certificates, you must manually configure each IPSec peer for every peer with which it communicates, and every new peer you add to a network would thus require a configuration change on every peer with which you need it to communicate securely.

When you use digital certificates, each peer is enrolled with a CA. When two peers attempt to communicate, they exchange certificates and digitally sign data to authenticate each other. When a new peer is added to the network, you enroll that peer with a CA and none of the other peers need modification. When the new peer attempts an IPSec connection, certificates are automatically exchanged and the peer can be authenticated.

With a CA, a peer authenticates itself to the remote peer by sending a certificate to the remote peer and performing some public key cryptography. Each peer sends its unique certificate which was issued by the CA. This process works because each certificate encapsulates the public key for the associated peer and each certificate is authenticated by the CA, and all participating peers recognize the CA as an authenticating authority. This is called IKE with an RSA signature.

The peer can continue sending its certificate for multiple IPSec sessions, and to multiple IPSec peers, until the certificate expires. When its certificate expires, the peer administrator must obtain a new one from the CA.

CAs can also revoke certificates for peers that no longer participate in IPSec. Revoked certificates are not recognized as valid by other peers. Revoked certificates are listed in a CRL, which each peer may check before accepting a certificate from another peer.

Some CAs have an RA as part of their implementation. An RA is a server that acts as a proxy for the CA so that CA functions can continue when the CA is unavailable.

About Key Pairs

Key pairs are RSA keys.

- RSA keys can be used for SSH or SSL.
- SCEP enrollment supports the certification of RSA keys.
- For the purposes of generating keys, the maximum key modulus for RSA keys is 2048 bits. The default size is 1024. Many SSL connections using identity certificates with RSA key pairs that exceed 1024 bits can cause a high CPU usage on the security appliance and rejected clientless logins.
- For signature operations, the supported maximum key size is 4096 bits.
- You can generate a general purpose RSA key pair, used for both signing and encryption, or you can generate separate RSA key pairs for each purpose.

Separate signing and encryption keys helps reduce exposure of the keys. This is because SSL uses a key for encryption but not signing but IKE uses a key for signing but not encryption. By using separate keys for each, exposure of the keys is minimized.

About Trustpoints

Trustpoints let you manage and track CAs and certificates. A trustpoint is a representation of a CA or identity pair. A trustpoint contains the identity of the CA, CA-specific configuration parameters, and an association with one enrolled identity certificate.

After you have defined a trustpoint, you can reference it by name in commands requiring that you specify a CA. You can configure many trustpoints.

Note If a security appliance has multiple trustpoints that share the same CA, only one of these trustpoints sharing the CA can be used to validate user certificates. Use the **support-user-cert-validation** command to control which trustpoint sharing a CA is used for validation of user certificates issued by that CA.

For automatic enrollment, a trustpoint must be configured with an enrollment URL and the CA that the trustpoint represents must be available on the network and must support SCEP.

You can export and import the keypair and issued certificates associated with a trustpoint in PKCS12 format. This is useful if you wish to manually duplicate a trustpoint configuration on a different security appliance.

About Revocation Checking

When a certificate is issued, it is valid for a fixed period of time. Sometimes a CA revokes a certificate before this time period expires; for example, due to security concerns or a change of name or association. CAs periodically issue a signed list of revoked certificates. Enabling revocation checking forces the security appliance to check that the CA has not revoked a certificate every time it uses that certificate for authentication.

When you enable revocation checking, during the PKI certificate validation process the security appliance checks certificate revocation status. It can use either CRL checking or Online Certificate Status Protocol or both, with the second method you set in effect only when the first method returns an error, for example, that the server is unavailable.

With CRL checking, the security appliance retrieves, parses, and caches Certificate Revocation Lists, which provide a complete list of revoked certificates. OCSP offers a more scalable method of checking revocation status in that it localizes certificate status on a Validation Authority, which it queries for the status of a specific certificate.

About CRLs

Certificate Revocation Lists provide the security appliance with one means of determining whether a certificate that is within its valid time range has been revoked by its issuing CA. CRL configuration is a part of the configuration of a trustpoint.

You can configure the security appliance to make CRL checks mandatory when authenticating a certificate (**revocation-check crl** command). You can also make the CRL check optional by adding the **none** argument (**revocation-check crl none** command), which allows the certificate authentication to succeed when the CA is unavailable to provide updated CRL data.

The security appliance can retrieve CRLs from CAs using HTTP, SCEP, or LDAP. CRLs retrieved for each trustpoint are cached for a length of time configurable for each trustpoint.

When the security appliance has cached a CRL for more than the length of time it is configured to cache CRLs, the security appliance considers the CRL too old to be reliable, or "stale". The security appliance attempts to retrieve a newer version of the CRL the next time a certificate authentication requires checking the stale CRL.

The security appliance caches CRLs for a length of time determined by the following two factors:

- The number of minutes specified with the **cache-time** command. The default value is 60 minutes.
- The NextUpdate field in the CRLs retrieved, which may be absent from CRLs. You control whether the security appliance requires and uses the NextUpdate field with the **enforcenextupdate** command.

The security appliance uses these two factors as follows:

- If the NextUpdate field is not required, the security appliance marks CRLs as stale after the length of time defined by the **cache-time** command.
- If the NextUpdate field is required, the security appliance marks CRLs as stale at the sooner of the two times specified by the **cache-time** command and the NextUpdate field. For example, if the cache-time command is set to 100 minutes and the NextUpdate field specifies that the next update is 70 minutes away, the security appliance marks CRLs as stale in 70 minutes.

If the security appliance has insufficient memory to store all CRLs cached for a given trustpoint, it deletes the least recently used CRL to make room for a newly retrieved CRL.

For information about configuring CRL behavior for a trustpoint, see the "Configuring CRLs for a Trustpoint" section on page 41-13.

About OCSP

Online Certificate Status Protocol provides the security appliance with a means of determining whether a certificate that is within its valid time range has been revoked by its issuing CA. OCSP configuration is a part of the configuration of a trustpoint.

OCSP localizes certificate status on a Validation Authority (an OCSP server, also called the *responder*) which the security appliance queries for the status of a specific certificate. It provides better scalability and more up-to-date revocation status than does CRL checking. It helps organizations with large PKI installations deploy and expand secure networks.

You can configure the security appliance to make OCSP checks mandatory when authenticating a certificate (**revocation-check ocsp** command). You can also make the OCSP check optional by adding the **none** argument (**revocation-check ocsp none** command), which allows the certificate authentication to succeed when the Validation Authority is unavailable to provide updated OCSP data.

Our implementation of OCSp provides three ways to define the OCSP server URL. The security appliance uses these servers in the following order:

1. The OCSP URL defined in a match certificate override rule (**match certificate** command).
2. The OCSP URL configured in the **ocsp url** command.
3. The AIA field of the client certificate.

Note To configure a trustpoint to validate a self-signed OCSP responder certificate, you import the self-signed responder certificate into its own trustpoint as a trusted CA certificate. Then you configure the **match certificate** command in the client certificate validating trustpoint to use the trustpoint that contains the

self-signed OCSP responder certificate to validate the responder certificate. The same applies for configuring validating responder certificates external to the validation path of the client certificate.

The OCSP server (responder) certificate typically signs the OCSP response. After receiving the response, the security appliance tries to verify the responder certificate. The CA normally sets the lifetime of its OCSP responder certificate to a relatively short period to minimize the chance of it being compromised.The CA typically also includes an ocsp-no-check extension in the responder certificate indicating that this certificate does not need revocation status checking. But if this extension is not present, the security appliance tries to check its revocation status using the same method specified in the trustpoint. If the responder certificate is not verifiable, revocation checks fails. To avoid this possibility, configure **revocation-check none** in the responder certificate validating trustpoint, while configuring r**evocation-check ocsp** for the client certificate.

Supported CA Servers

The security appliance supports the following CA servers:

- Cisco IOS CS
- Baltimore Technologies
- Entrust
- Microsoft Certificate Services
- Netscape CMS
- RSA Keon
- VeriSign

Certificate Configuration

This section describes how to configure the security appliance with certificates and other procedures related to certificate use and management.

This section includes the following topics:

- Preparing for Certificates, page 41-5
- Configuring Key Pairs, page 41-6
- Configuring Trustpoints, page 41-7
- Obtaining Certificates, page 41-9
- Configuring CRLs for a Trustpoint, page 41-13
- Exporting and Importing Trustpoints, page 41-15
- Configuring CA Certificate Map Rules, page 41-16

Preparing for Certificates

Before you configure a security appliance with certificates, ensure that the security appliance is configured properly to support certificates. An improperly configured security appliance can cause enrollment to fail or for enrollment to request a certificate containing inaccurate information.

To prepare a security appliance for certificates, perform the following steps:

Step 1 Ensure that the hostname and domain name of the security appliance are configured correctly. You can use the **show running-config** command to view the hostname and domain name as currently configured.

For information about configuring the hostname, see the "Setting the Hostname" section on page 9-2.

For information about configuring the domain name, see the "Setting the Domain Name" section on page 9-2.

Step 2 Be sure that the security appliance clock is set accurately before configuring the CA. Certificates have a date and time that they become valid and that they expire. When the security appliance enrolls with a CA and gets a certificate, the security appliance checks that the current time is within the valid range for the certificate. If it is outside that range, enrollment fails.

For information about setting the clock, see the "Setting the Date and Time" section on page 9-2.

Configuring Key Pairs

This section includes the following topics:

- Generating Key Pairs, page 41-6
- Removing Key Pairs, page 41-7

Generating Key Pairs

Key pairs are RSA keys, as discussed in the "About Key Pairs" section on page 41-2. You must generate key pairs for the types of certification you want to use.

To generate key pairs, perform the following steps:

Step 1 Generate the types of key pairs needed for your PKI implementation. To do so, perform the following steps, as applicable:

a. If you want to generate RSA key pairs, use the **crypto key generate rsa** command.

```
hostname/contexta(config)# crypto key generate rsa
```

If you do not use additional keywords this command generates one general purpose RSA key pair. Because the key modulus is not specified, the default key modulus of 1024 is used. You can specify other modulus sizes with the **modulus** keyword.

Note Many SSL connections using identity certificates with RSA key pairs that exceed 1024 bits can cause a high CPU usage on the security appliance and rejected clientless logins.

You can also assign a label to each key pair using the **label** keyword. The label is referenced by the trustpoint that uses the key pair. If you do not assign a label, the key pair is automatically labeled <Default-RSA-Key>.

```
hostname/contexta(config)# crypto key generate rsa label key-pair-label
```

Step 2 (Optional) Use the **show crypto key mypubkey** command to view key pair(s). The following example shows an RSA general-purpose key:

```
hostname/contexta(config)# show crypto key mypubkey
Key pair was generated at: 16:39:47 central Feb 10 2005
Key name: <Default-RSA-Key>
 Usage: General Purpose Key
 Modulus Size (bits): 1024
 Key Data:

  30819f30 0d06092a 864886f7 0d010101 05000381 8d003081 89028181 00ea51b7
  0781848f 78bccac2 4a1b5b8d 2f3e30b4 4cae9f86 f4485207 159108c9 f5e49103
  9eeb0f5d 45fd1811 3b4aafce 292b3b64 b4124a6f 7a777b08 75b88df1 8092a9f8
  5508e9e5 2c271245 7fd1c0c3 3aaf1e04 c7c4efa4 600f4c4a 6afe56ad c1d2c01c
  e08407dd 45d9e36e 8cc0bfef 14f9e6ac eca141e4 276d7358 f7f50d13 79020301 0001
Key pair was generated at: 16:34:54 central Feb 10 2005
```

Step 3 Save the key pair you have generated. To do so, save the running configuration by entering the **write memory** command.

Removing Key Pairs

To remove key pairs, use the **crypto key zeroize command in global configuration mode.**

The following example removes RSA key pairs:

```
hostname(config)# crypto key zeroize rsa
WARNING: All RSA keys will be removed.
WARNING: All device certs issued using these keys will also be removed.

Do you really want to remove these keys? [yes/no] y
hostname(config)#
```

Configuring Trustpoints

For information about trustpoints, see the "About Trustpoints" section on page 41-3.

To configure a trustpoint, perform the following steps:

Step 1 Create a trustpoint corresponding to the CA from which the security appliance needs to receive its certificate.

```
hostname/contexta(config)# crypto ca trustpoint trustpoint
```

For example, to declare a trustpoint called Main:

```
hostname/contexta(config)# crypto ca trustpoint Main
hostname/contexta(config-ca-trustpoint)#
```

Upon entering this command, you enter the Crypto ca trustpoint configuration mode.

Step 2 Specify the enrollment method to be used with this trustpoint.

To specify the enrollment method, do one of the following items:

- To specify SCEP enrollment, use the **enrollment url** command to configure the URL to be used for SCEP enrollment with the trustpoint you declared. For example, if the security appliance requests certificates from trustpoint Main using the URL http://10.29.67.142:80/certsrv/mscep/mscep.dll, then the command would be as follows:

```
hostname/contexta(config-ca-trustpoint)# enrollment url
http://10.29.67.142:80/certsrv/mscep/mscep.dll
```

- To specify manual enrollment, use the **enrollment terminal** command to indicate that you will paste the certificate received from the CA into the terminal.

Step 3 As needed, specify other characteristics for the trustpoint. The characteristics you need to define depend upon your CA and its configuration. You can specify characteristics for the trustpoint using the following commands. Refer to the *Cisco Security Appliance Command Reference* for complete descriptions and usage guidelines of these commands.

- **accept-subordinates**—Indicates whether CA certificates subordinate to the CA associated with the trustpoint are accepted if delivered during phase one IKE exchange when not previously installed on the device.
- **crl required | optional | nocheck**—Specifies CRL configuration options. When you enter the **crl** command with the **optional** keyword included within the command statement, certificates from peers can still be accepted by your security appliance even if the CRL is not accessible to your security appliance.

  Note If you chose to enable required or optional CRL checking, be sure you configure the trustpoint for CRL managemen2t, which should be completed after you have obtained certificates. For details about configuring CRL management for a trustpoint, see the "Configuring CRLs for a Trustpoint" section on page 41-13.

- **crl configure**—Enters CRL configuration mode.
- **default enrollment**—Returns all enrollment parameters to their system default values. Invocations of this command do not become part of the active configuration.
- **email** *address*—During enrollment, asks the CA to include the specified email address in the Subject Alternative Name extension of the certificate.
- **enrollment retry period** —(Optional) Specifies a retry period in minutes. This characteristic only applies if you are using SCEP enrollment.
- **enrollment retry count**—(Optional) Specifies a maximum number of permitted retries. This characteristic only applies if you are using SCEP enrollment.
- **enrollment terminal**—Specifies cut and paste enrollment with this trustpoint.
- **enrollment url** *URL*—Specifies automatic enrollment (SCEP) to enroll with this trustpoint and configures the enrollment URL.
- **fqdn** *fqdn*—During enrollment, asks the CA to include the specified fully qualified domain name in the Subject Alternative Name extension of the certificate.
- **id-cert-issuer**—Indicates whether the system accepts peer certificates issued by the CA associated with this trustpoint.
- **ip-address** *ip-address*—During enrollment, asks the CA to include the IP address of the security appliance in the certificate.
- **keypair** *name*—Specifies the key pair whose public key is to be certified.
- **match certificate map**—Configures OCSP URL overrides and trustpoints to use to validate OCSP responder certificates
- **ocsp disable-nonce**—Disable the nonce extension on an OCSP request; the nonce extension cryptographically binds requests with responses to avoid replay attacks.

- **ocsp url**—Configures an OCSP server for the security appliance to use to check all certificates associated with a trustpoint rather than the server specified in the AIA extension of the client certificate.
- **password** *string*—Specifies a challenge phrase that is registered with the CA during enrollment. The CA typically uses this phrase to authenticate a subsequent revocation request.
- **revocation-check**—Sets one or more methods for revocation checking: CRL, OCSP, and none.
- **subject-name** *X.500 name*—During enrollment, asks the CA to include the specified subject DN in the certificate. If a DN string contains a comma, enclose the value string with double quotes (for example, O="Company, Inc.").
- **serial-number**—During enrollment, asks the CA to include the security appliance serial number in the certificate.
- **support-user-cert-validation**—If enabled, the configuration settings to validate a remote user certificate can be taken from this trustpoint, provided that this trustpoint is authenticated to the CA that issued the remote certificate.
- **exit**—Leaves the mode.

Step 4 Save the trustpoint configuration. To do so, save the running configuration by entering the **write memory** command.

Obtaining Certificates

The security appliance needs a CA certificate for each trustpoint and one or two certificates for itself, depending upon the configuration of the keys used by the trustpoint. If the trustpoint uses separate RSA keys for signing and encryption, the security appliance needs two certificates, one for each purpose. In other key configurations, only one certificate is needed.

The security appliance supports enrollment with SCEP and with manual enrollment, which lets you paste a base-64-encoded certificate directly into the terminal. For site-to-site VPNs, you must enroll each security appliance. For remote access VPNs, you must enroll each security appliance and each remote access VPN client.

This section includes the following topics:

- Obtaining Certificates with SCEP, page 41-9
- Obtaining Certificates Manually, page 41-11
- Using Extended Keys for Certificates, page 41-13

Obtaining Certificates with SCEP

This procedure provides steps for configuring certificates using SCEP. Repeat these steps for each trustpoint you configure for automatic enrollment. When you have completed this procedure, the security appliance will have received a CA certificate for the trustpoint and one or two certificates for signing and encryption purposes. If you use general-purpose RSA keys, the certificate received is for signing and encryption. If you use separate RSA keys for signing and encryption, the security appliance receives separate certificates for each purpose.

Note Whether a trustpoint uses SCEP for obtaining certificates is determined by the use of the **enrollment url** command when you configure the trustpoint (see the "Configuring Trustpoints" section on page 41-7).

To obtain certificates with SCEP, perform the following steps:

Step 1 Obtain the CA certificate for the trustpoint you configured.

```
hostname/contexta(config)# crypto ca authenticate trustpoint
```

For example, using trustpoint named Main, which represents a subordinate CA:

```
hostname/contexta(config)# crypto ca authenticate Main

INFO: Certificate has the following attributes:
Fingerprint:     3736ffc2 243ecf05 0c40f2fa 26820675
Do you accept this certificate? [yes/no]: y

Trustpoint 'Main' is a subordinate CA and holds a non self signed cert.
Trustpoint CA certificate accepted.
```

Step 2 Enroll the security appliance with the trustpoint. This process retrieves a certificate for signing data and, depending upon the type of keys you configured, for encrypting data.

Step 3 To perform enrollment, use the **crypto ca enroll** command. Before entering this command, contact your CA administrator because the administrator may need to authenticate your enrollment request manually before the CA grants its certificates.

```
hostname(config)# crypto ca enroll trustpoint
```

If the security appliance does not receive a certificate from the CA within 1 minute (the default) of sending a certificate request, it resends the certificate request. The security appliance continues sending a certificate request every 1 minute until a certificate is received.

Note If the fully qualified domain name configured for the trustpoint is not identical to the fully qualified domain name of the security appliance, including the case of the characters, a warning appears. If needed, you can exit the enrollment process, make any necessary corrections, and enter the **crypto ca enroll** command again.

The following enrollment example performs enrollment with the trustpoint named Main:

```
hostname(config)# crypto ca enroll Main
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
% password to the CA Administrator in order to revoke your certificate.
% For security reasons your password will not be saved in the configuration.
% Please make a note of it.
Password: 2b0rn0t2b
Re-enter password: 2b0rn0t2b
% The subject name in the certificate will be: securityappliance.example.com
% The fully-qualified domain name in the certificate will be:
securityappliance.example.com
% Include the device serial number in the subject name? [yes/no]: no
Request certificate from CA [yes/no]: yes
% Certificate request sent to Certificate authority.
```

Note The password is required if the certificate for the security appliance needs to be revoked, so it is crucial that you remember this password. Note it and store it in a safe place.

You must enter the **crypto ca enroll** command for each trustpoint with which the security appliance needs to enroll.

Note If your security appliance reboots after you issued the **crypto ca enroll** command but before you received the certificate, reissue the **crypto ca enroll** command and notify the CA administrator.

Step 4 Verify that the enrollment process was successful using the **show crypto ca certificate** command. For example, to show the certificate received from trustpoint Main:

```
hostname/contexta(config)# show crypto ca certificate Main
```

The output of this command shows the details of the certificate issued for the security appliance and the CA certificate for the trustpoint.

Step 5 Save the configuration using the **write memory** command:

```
hostname/contexta(config)# write memory
```

Obtaining Certificates Manually

This procedure provides steps for configuring certificates using manual certificate requests. Repeat these steps for each trustpoint you configure for manual enrollment. When you have completed this procedure, the security appliance will have received a CA certificate for the trustpoint and one or two certificates for signing and encryption purposes. If you use general-purpose RSA keys, the certificate received is for signing and encryption. If you use separate RSA keys for signing and encryption, the certificates received are used for each purpose exclusively.

Note Whether a trustpoint requires that you manually obtain certificates is determined by the use of the **enrollment terminal** command when you configure the trustpoint (see the "Configuring Trustpoints" section on page 41-7).

To obtain certificates manually, perform the following steps:

Step 1 Obtain a base-64 encoded CA certificate from the CA represented by the trustpoint.

Step 2 Import the CA certificate. To do so, use the **crypto ca authenticate** command. The following example shows a CA certificate request for the trustpoint Main.

```
hostname (config)# crypto ca authenticate Main
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
MIIDRTCCAu+gAwIBAgIQKVcqP/KW74VP0NZzL+JbRTANBgkqhkiG9w0BAQUFADCB
[ certificate data omitted ]
/7QEM8izy0EOTSErKu7Nd76jwf5e4qttkQ==
quit

INFO: Certificate has the following attributes:
Fingerprint:     24b81433 409b3fd5 e5431699 8d490d34
```

```
Do you accept this certificate? [yes/no]: y
Trustpoint CA certificate accepted.

% Certificate successfully imported
hostname (config)#
```

Step 3 Generate a certificate request. To do so, use the **crypto ca enroll** command. The following example shows a certificate and encryption key request for the trustpoint Main, which is configured to use manual enrollment and general-purpose RSA keys for signing and encryption.

```
hostname (config)# crypto ca enroll Main
% Start certificate enrollment ..

% The fully-qualified domain name in the certificate will be:
securityappliance.example.com

% Include the device serial number in the subject name? [yes/no]: n

Display Certificate Request to terminal? [yes/no]: y
Certificate Request follows:

MIIBoDCCAQkCAQAwIzEhMB8GCSqGSIb3DQEJAhYSRmVyYWxQaXguY2lzY28uY29t
[ certificate request data omitted ]
jF4waw68eOxQxVmdgMWeQ+RbIOYmvt8g6hnBTrd0GdqjjVLt

---End - This line not part of the certificate request---

Redisplay enrollment request? [yes/no]: n
hostname (config)#
```

Note If you use separate RSA keys for signing and encryption, the **crypto ca enroll** command displays two certificate requests, one for each key. To complete enrollment, acquire a certificate for all certificate requests generated by the **crypto ca enroll** command.

Step 4 For each request generated by the **crypto ca enroll** command, obtain a certificate from the CA represented by the applicable trustpoint. Be sure the certificate is in base-64 format.

Step 5 For each certificate you receive from the CA, use the **crypto ca import certificate** command. The security appliance prompts you to paste the certificate to the terminal in base-64 format.

Note If you use separate RSA key pairs for signing and encryption, perform this step for each certificate separately. The security appliance determines automatically whether the certificate is for the signing or encryption key pair. The order in which you import the two certificates is irrelevant.

The following example manually imports a certificate for the trustpoint Main:

```
hostname (config)# crypto ca import Main certificate
% The fully-qualified domain name in the certificate will be:
securityappliance.example.com

Enter the base 64 encoded certificate.
End with a blank line or the word "quit" on a line by itself
[ certificate data omitted ]
quit
INFO: Certificate successfully imported
hostname (config)#
```

Step 6 Verify that the enrollment process was successful using the **show crypto ca certificate** command. For example, to show the certificate received from trustpoint Main:

```
hostname/contexta(config)# show crypto ca certificate Main
```

The output of this command shows the details of the certificate issued for the security appliance and the CA certificate for the trustpoint.

Step 7 Save the configuration using the **write memory** command:

```
hostname/contexta(config)# write memory
```

Using Extended Keys for Certificates

After you have upgraded the security appliance (for example, from Version 7.2 to Version 8.0), you can connect to it with the Cisco AnyConnect VPN Client using an older certificate with an entended key OID. To allow you to migrate older certificates, two new submode options to the **crypto ca trustpoint** command have been added. To enable this feature, perform the following steps:

Step 1 Enter the **crypto ca trustpoint** command to enter the crypto ca trustpoint configuration mode:

```
hostname (config)# crypto ca trustpoint trustpoint-name
hostname(config-ca-trustpoint)
```

Step 2 To perform key usage checks for an SSL certificate, enter the followingsubmode option:

```
hostname(config-ca-trustpoint) no ignore-ssl-keyusage
```

Step 3 To perform key usage checks for an IPSec certificate, enter the following submode option:

```
hostname(config-ca-trustpoint) no ignore-ipsec-keyusage
```

Configuring CRLs for a Trustpoint

If you want to use mandatory or optional CRL checking during certificate authentication, you must perform CRL configuration for each trustpoint. For more information about CRLs, see the "About CRLs" section on page 41-3.

To configure CRLs for a trustpoint, perform the following steps:

Step 1 Enter crypto ca trustpoint configuration mode for the trustpoint whose CRL configuration you want to modify. To do so, enter the **crypto ca trustpoint** command.

Step 2 If you have not already enabled CRLs, you can do so now by using the **crl** command with either the **required** or **optional** keyword. If you specify the **required** keyword, certificate authentication with this trustpoint cannot succeed if the CRL is unavailable.

Step 3 Enter the **crl configure** command.

```
hostname/contexta(config-ca-trustpoint)# crl configure
hostname/contexta(config-ca-crl)#
```

Upon entering this command, you enter the crl configuration mode for the current trustpoint.

Tip To set all CRL configuration options to their default values, use the **default** command. At any time while performing CRL configuration, if you want to start over, enter this command and restart this procedure.

Step 4 Configure the retrieval policy with the **policy** command. The following keywords for this command determine the policy.

- **cdp**—CRLs are retrieved only from the CRL distribution points specified in authenticated certificates.

  Note SCEP retrieval is not supported by distribution points specified in certificates.

- **static**—CRLs are retrieved only from URLs you configure.
- **both**—CRLs are retrieved from CRL distribution points specified in authenticated certificates and from URLs you configure.

Step 5 If you used the keywords static or both when you configured the CRL policy, you need to configure URLs for CRL retrieval, using the **url** command. You can enter up to 5 URLs, ranked 1 through 5.

```
hostname/contexta(config-ca-crl)# url n URL
```

where *n* is the rank assigned to the URL. To remove a URL, use the **no url** *n* command.

Step 6 Configure the retrieval method with the **protocol** command. The following keywords for this command determine the retrieval method.

- **http**—Specifies HTTP as the CRL retrieval method.
- **ldap**—Specifies LDAP as the CRL retrieval method.
- **scep**—Specifies SCEP as the CRL retrieval method.

Step 7 Configure how long the security appliance caches CRLs for the current trustpoint. To specify the number of minutes the security appliance waits before considering a CRL stale, enter the following command.

```
hostname/contexta(config-ca-crl)# cache-time n
```

where *n* is the number of minutes. For example, to specify that CRLs should be cached for seven hours, enter the following command.

```
hostname/contexta(config-ca-crl)# cache-time 420
```

Step 8 Configure whether the security appliance requires the NextUpdate field in CRLs. For more information about how the security appliance uses the NextUpdate field, see the "About CRLs" section on page 41-3.

Do one of the following:

- To require the NextUpdate field, enter the **enforcenextupdate** command. This is the default setting.
- To allow the NextUpdate field to be absent in CRLs, enter the **no enforcenextupdate** command.

Step 9 If you specified LDAP as the retrieval protocol, perform the following steps:

a. Enter the following command to identify the LDAP server to the security appliance:

```
hostname/contexta(config-ca-crl)# ldap-defaults server
```

You can specify the server by DNS hostname or by IP address. You can also provide a port number if the server listens for LDAP queries on a port other than the default of 389. For example, the following command configures the security appliance to retrieve CRLs from an LDAP server whose hostname is ldap1.

```
hostname/contexta(config-ca-crl)# ldap-defaults ldap1
```

Note If you use a hostname rather than an IP address to specify the LDAP server, be sure you have configured the security appliance to use DNS. For information about configuring DNS, see the **dns** commands in the *Cisco Security Appliance Command Reference*.

b. If LDAP server requires credentials to permit CRL retrieval, enter the following command:

```
hostname/contexta(config-ca-crl)# ldap-dn admin-DN password
```

For example:

```
hostname/contexta(config-ca-crl)# ldap-dn cn=admin,ou=devtest,o=engineering c00lRunZ
```

Step 10 To test CRL configuration for the current trustpoint, use the **crypto ca crl request** command. This command retrieves the current CRL from the CA represented by the trustpoint you specify.

Step 11 Save the running configuration. Enter the **write memory** command.

Exporting and Importing Trustpoints

You can export and import keypairs and issued certificates associated with a trustpoint configuration. The security appliance supports PKCS12 format for the export and import of trustpoints.

This section includes the following topics:

- Exporting a Trustpoint Configuration, page 41-15
- Importing a Trustpoint Configuration, page 41-16

Exporting a Trustpoint Configuration

To export a trustpoint configuration with all associated keys and certificates in PKCS12 format, use the **crypto ca export** command. The security appliance displays the PKCS12 data in the terminal. You can copy the data. The trustpoint data is password protected; however, if you save the trustpoint data in a file, be sure the file is in a secure location.

The following example exports PKCS12 data for trustpoint Main using Wh0zits as the passphrase:

```
hostname (config)# crypto ca export Main pkcs12 Wh0zits

Exported pkcs12 follows:

[ PKCS12 data omitted ]

---End - This line not part of the pkcs12---

hostname (config)#
```

Importing a Trustpoint Configuration

To import the keypairs and issued certificates associated with a trustpoint configuration, use the **crypto ca import pkcs12** command in global configuration mode. The security appliance prompts you to paste the text to the terminal in base-64 format.

The key pair imported with the trustpoint is assigned a label matching the name of the trustpoint you create. For example, if an exported trustpoint used an RSA key labeled <Default-RSA-Key>, creating trustpoint named Main by importing the PKCS12 creates a key pair named Main, not <Default-RSA-Key>.

Note If a security appliance has trustpoints that share the same CA, only one of the trustpoints sharing the CA can be used to validate user certificates. The **crypto ca import pkcs12** command can create this situation. Use the **support-user-cert-validation** command to control which trustpoint sharing a CA is used for validation of user certificates issued by that CA.

The following example manually imports PKCS12 data to the trustpoint Main with the passphrase Wh0zits:

```
hostname (config)# crypto ca import Main pkcs12 Wh0zits

Enter the base 64 encoded pkcs12.
End with a blank line or the word "quit" on a line by itself:
[ PKCS12 data omitted ]
quit
INFO: Import PKCS12 operation completed successfully
hostname (config)#
```

Configuring CA Certificate Map Rules

You can configure rules based on the Issuer and Subject fields of a certificate. Using the rules you create, you can map IPSec peer certificates to tunnel groups with the **tunnel-group-map** command. The security appliance supports one CA certificate map, which can contain many rules. For more information about using CA certificate map rules with tunnel groups, see the "Creating a Certificate Group Matching Rule and Policy" section on page 29-9.

To configure a CA certificate map rule, perform the following steps:

Step 1 Enter CA certificate map configuration mode for the rule you want to configure. To do so, enter the **crypto ca certificate map command and specify the rule index number.** The following example enters CA certificate map mode for the rule with index number 1.

```
hostname(config)# crypto ca certificate map 1
hostname(config-ca-cert-map)#
```

Step 2 Use the **issuer-name** and **subject-name** commands to configure the rule. These commands specify tests that the security appliance can apply to values found in the Issuer or Subject fields of certificates. The tests can apply to specific attributes or to the whole of the Issuer or Subject fields. You can configure many tests per rule, and all the tests you specify with these commands must be true for a rule to match a certificate. Valid operators in the **issuer-name** and **subject-name** commands are as follows.

Operator	Meaning
eq	The field or attribute must be identical to the value given.
ne	The field or attribute cannot be identical to the value given.
co	Part or all of the field or attribute must match the value given.
nc	No part of the field or attribute can match the value given.

For more information about the **issuer-name** and **subject-name** commands, see the *Cisco Security Appliance Command Reference*.

The following example specifies that any attribute within the Issuer field must contain the string ASC:

```
hostname(config-ca-cert-map)# issuer-name co asc
hostname(config-ca-cert-map)#
```

The following example specifies that within the Subject field an Organizational Unit attribute must exactly match the string Engineering.

```
hostname(config-ca-cert-map)# subject-name attr ou eq Engineering
hostname(config-ca-cert-map)#
```

Map rules appear in the output of the **show running-config** command.

```
crypto ca certificate map 1
 issuer-name co asc
 subject-name attr ou eq Engineering
```

Step 3 When you have finished configuring the map rule, save your work. Enter the **write memory** command.

The Local CA

The Local Certificate Authority (Local CA) performs the following tasks:

- Integrates basic certificate authority functionality on the security appliance.
- Deploys certificates.
- Provides secure revocation checking of issued certificates.
- Provides a certificate authority on the adaptive security appliance for use with SSL VPN connections, both browser- and client-based.
- Provides trusted digital certificates to users, without the need to rely on external certificate authorization.
- Provides a secure in-house authority for certificate authentication and offers straightforward user enrollment by means of a browser web page login.

Note The Local CA feature is not supported if you use active/active failover or VPN load balancing. The Local CA cannot be subordinate to another CA; it can act only as the Root CA.

After you configure a Local CA server on the security appliance, users can enroll for a certificate by visiting a specified browser-based enrollment page and entering a username and a one-time password that is provided by the Local CA administrator to validate their eligibility for enrollment.

As shown in Figure 41-1, the Local CA server, configurable from both the CLI and ASDM, resides on the security appliance and handles enrollment requests from web page users and CRL inquiries coming from other certificate validating devices and security appliances. Local CA database and configuration files are maintained either on the security appliance flash memory (default storage) or on a separate storage device.

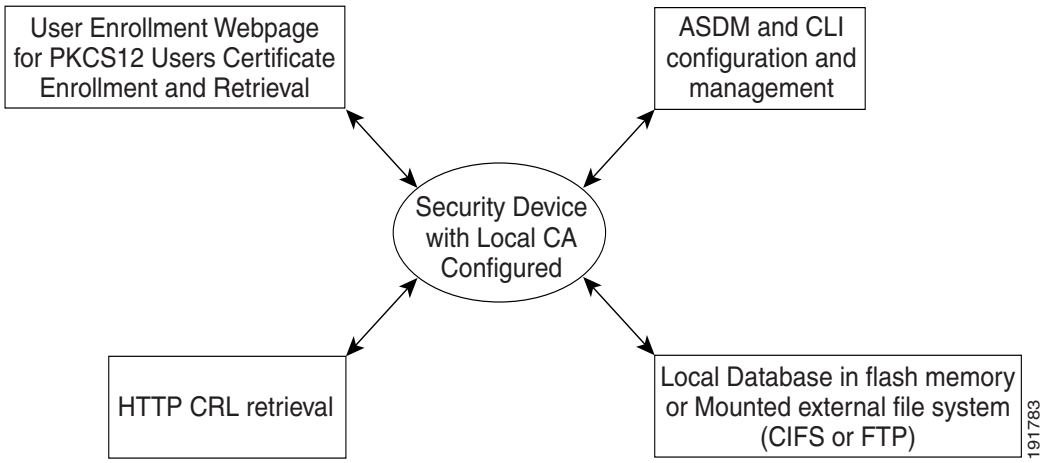


Figure 41-1 The Local Certificate Authority (CA)

Note Only one Local CA server can be resident on a security appliance at a time, and the Local CA cannot be configured as a subordinate to an external CA.

Configuring the Local CA Server

This section describes how to configure the Local CA server on the security appliance and includes the following topics:

- The Default Local CA Server, page 41-19
- Customizing the Local CA Server, page 41-20
- Certificate Characteristics, page 41-21

The Default Local CA Server

The default Local CA server requires only a few configuration commands to set up with the following characteristics. Once you use the **crypto ca server** command to access config-ca-server mode, all you must specify are CLI commands described in the following steps:

Step 1 Specify the SMTP (Simple Mail Transfer Protocol) from-address with the **smtp from-address** command. This command provides a valid e-mail address the Local CA uses as a from: address when sending e-mails that deliver one-time passwords for an enrollment invitation to users.

Step 2 For an optional subject-name DN appended to each username on issued certificates, specify the subject-name DN with the **subject-name-default** command. The subject-name DN and the username combine to form the DN in all user certificates issued by the Local CA server. If you do not specify a subject-name DN, you must specify the exact subject name DN to be included in a user certificate each time you add a user to the user database.

The following example shows the few CLI commands required to configure and enable the Local CA server when you are using the predefined default values for all required parameters.

```
hostname(config)# crypto ca server
hostname (config-ca-server) # smtp from-address SecurityAdmin@hostcorp.com
hostname (config-ca-server)# subject-name-default cn=engineer, o=asc Systems, c=US
hostname(config-ca-server)# no shutdown
```

All other required parameter values are the system defaults. Table 41-1 lists the configurable characteristics of the Local CA server, their pre-defined default values, and the CLI commands that configure them.

Note **Issuer-name** and **keysize server** values cannot be changed after you enable the Local CA initially. Be sure to review all optional parameters carefully before you enable the configured Local CA.

Table 41-1 Local CA Local CA Server Default Characteristics

Local CA Server Characteristic	Default Value	CLI Configuration Command(s)
Storage Location for database and configuration	On-board flash memory in the directory LOCAL-CA-SERVER.	**mount** `(global config mode)` **database path**
Certificate Issuer Name	cn=*FQDN*	**issuer-name**
Enabled/disabled. **no-shutdown** enables the Local CA; **shutdown** disables it.	No Local CA Server configured.	**shutdown vs. no shutdown** (enables)
Access to config-ca-server mode and Local CA server configuration commands	No server enabled	**crypto ca server**
Issued certificate keypair size	1024 bits per key	**keysize**
Local CA Certificate key-pair size	1024 bits per key	**keysize server**
Length of time a user certificate, server certificate, or CRL is valid	User Certificate=1 yr.; Server Certificate=3 yrs.; CRL=6 hours	**lifetime**
Length of time a one-time password is valid	Expires in 72 hrs. (three days)	otp-expiration

Local CA Server Characteristic	Default Value	CLI Configuration Command(s)
Certificate Revocation List (CRL) Distribution Point (CDP), the location of the CRL on the Local CA security appliance or on an external server	For a local CRL, the same as security appliance, http://*hostname.domain*/+CSCOCA+/asa_ca.crl	cdp-url
* E-mail address issuing Local CA e-mail notices	**Required.** You must supply an e-mail address as the default, admin@*FQDN,* might not be an actual address.	smtp from-address
Subject line in Local CA e-mail notices	"Certificate Enrollment Invitation"	smtp subject
* subject-name DN default to append to a username on issued certificates	**Optional. No default.** Supply a subject-name default value.	subject-name-default
Days before expiration reminders are sent.	14 days prior to expiration	**renewal-reminder**
Post-enrollment/renewal period an issued certificate file is available for re-use.	24 hours	**enrollment-retrieval**

*Indicates values without defaults that you must configure.

Once the **crypto ca server** command executes, the Local CA is generated. A self-signed certificate is created and associated with that Local CA on the security appliance when you execute the **no shutdown** command. The self-signed certificate key usage extension has key encryption, key signature, CRL signing, and certificate signing ability.

You can debug the configured default Local CA server with the **debug crypto ca server** command, which displays debug messages during configuration and test. This command is detailed further on in the section, Enabling the Local CA Server.

Note Once the self-signed Local CA certificate is generated, to modify its characteristics you must delete the existing Local CA server and completely recreate it.

Customizing the Local CA Server

This section describes configuring and enabling the Local CA server. Enabling it for the first time generates the server certificate and keypair, which automatically produces a CA. To begin configuring the Local CA server you must be in config-ca-server mode.

Once you execute the **crypto ca server** command to enter config-ca-server mode, you can begin to configure the various parameters of the Local CA server on the security appliance. Typically, to configure a customized Local CA server on a security appliance, you would perform the following steps:

Step 1 Enter the **crypto ca server** command to access the Local CA Server Configuration mode CLI command set, which allows you to configure and manage a Local CA. An example follows:

```
hostname (config)# crypto ca server
hostname (config-ca-server)#
```

Step 2 As with the default Local CA server, you must specify the parameters that do not have defaults, specifically the issuer-name command. An example follows:

```
hostname (config-ca-server)# issuer-name
CN=xx5520,CN=30.132.0.25,ou=DevTest,ou=QA,O=ASC Systems
hostname (config-ca-server)#
```

Step 3 To customize the text that appears in the subject field of all e-mails sent from the Local CA server, use the **smtp subject** *subject-line* command as follows:

```
hostname (config-ca-server) # smtp subject Priority E-Mail: Enclosed Confidential
Information is Required for Enrollment
hostname (config-ca-server)#
```

Step 4 To specify the e-mail address that is to be used as the From: field of all e-mails generated by the Local CA server, use the **smtp from-address** command as follows:

```
hostname (config-ca-server) # smtp from-address SecurityAdmin@hostcorp.com
hostname (config-ca-server)#
```

Step 5 To specify an optional subject-name DN to be appended to a username on issued certificates, use the **subject-name-default** command. The default subject-name DN becomes part of the username in all user certificates issued by the Local CA server. For example, if the username is maryjane@ASC.com and you set the subject-name default to cn=engineer, o=ASC Systems, c=US, the subject-name DN in the certificate would be cn=maryjane@ASC.com, cn=Engineer, o=ASC Systems, c=US.

Note If you do not specify a subject-name-default to serve as a standard subject-name default, you must specify a DN each time you add a user.

The permitted DN attribute keywords are listed in the following table:

Subject-name-default Keywords	
CN= Common Name	C = Country
SN = Surname	OU = Organization Unit
T = Title	EA = E-mail Address
O = Organization Name	ST = State/Province
L = Locality	

An example follows:

```
hostname (config-ca-server) # subject-name-default cn=engineer, o=ABC Systems, c=US
hostname (config-ca-server)#
```

Note that there are additional Local CA server commands that allow you to customize your server further. These commands are described in the following sections.

Certificate Characteristics

Configurable Local CA certificate characteristics include the following:

- The name of the certificate issuer as it appears on all user certificates
- The lifetime of the Local CA certificates (server and user) and the CRL
- The length of the public and private keypair associated with Local CA and user certificates.

Issuer Name

The certificate issuer name that is configured is both the subject-name and issuer-name of the self-signed Local CA certificate, as well as the issuer-name in all client certificates that are issued and in the issued CRL. The default issuer name in the Local CA is *hostname.domainname*. Use the **issuer-name** command to specify the Local CA certificate subject-name as shown in the following example:

```
hostname(config-ca-server)# issuer-name CN=xx5520,CN=30.132.0.25,ou=DevTest,ou=QA,O=ABC
Systems
hostname(config-ca-server)#
```

Note **The issuer-name** value cannot be changed after the initial enabling of the Local CA.

CA Certificate Lifetime

You can specify the lifetime, the period of validity for the Local CA certificate, all issued user certificates, or the CRL with the **lifetime** command. This command determines the expiration date included in the certificate; the default lifetime of a Local CA certificate is three years.

Use the **lifetime ca-certificate** command to set the number of days that you want the Local CA server certificate to remain valid as shown in the following example of configuring a Local CA certificate to last for one year:

```
hostname (config)# crypto ca server
hostname (config-ca-server)# lifetime ca-certificate 365
hostname (config-ca-server)
```

To reset the Local CA certificate lifetime to the default of three years during configuration, use the **no lifetime ca-certificate command. You can u**se the same command (or its **no** form) to specify (or reset) the valid lifetime of user certificates (**lifetime certificate...)** and the CRL (**lifetime crl...**).

The Local CA Server automatically generates a replacement CA certificate 30 days prior to the CA certificate expiration, allowing the replacement certificate to be exported and imported onto any other devices for certificate validation of user certificates issued by the Local CA certificate after expiration of the current Local CA certificate. The pre-expiration Syslog message:

```
%ASA-1-717049: Local CA Server certificate is due to expire in <days> days and a replace-
ment certificate is available for export.
```

Note When notified of this automatic rollover, the administrator must take action to ensure the new Local CA certificate is imported to all necessary devices prior to expiration.

User Certificate Lifetime

To set the number of days that you want user certificates to remain valid, use the **lifetime certificate** command as shown in the following example of configuring all user certificates to be valid for two months:

```
hostname (config)# crypto ca server
hostname (config-ca-server)# lifetime certificate 60
hostname(config-ca-server)#
```

Prior to a user certificate expiring, the Local CA server automatically initiates certificate renewal processing by granting that user enrollment privileges a number of days ahead of the certificate expiration, renewal-reminder setting, and by delivering an e-mail with the enrollment username and OTP for renewal of the certificate.

CRL Lifetime

The Local CA updates and reissues the CRL every time a user certificate is revoked or unrevoked, but if there are no revocation changes, the CRL is reissued automatically once every CRL lifetime, the period of time you specify with the **lifetime crl** command during Local CA configuration. If you do not specify a CRL lifetime, the default time period is six hours.

Use the **lifetime crl** command to set the number of hours that you want the certificate revocation list to remain valid as shown in the following example:

```
hostname (config)# crypto ca server
hostname (config-ca-server)# lifetime crl 10
hostname (config-ca-server)#
```

To force the issuance of a CRL at any time, you can use the **crypto ca server crl issue** command, which immediately updates and regenerates a current CRL to overwrite the existing CRL. This command can force the issuance of a CRL in any circumstances, such as a corrupt or destroyed CRL file.

This command displays a message indicating that the CRL is updated. An example follows:

```
hostname(config)# crypto ca server crl issue
A new CRL has been issued.
hostname(config)#
```

Note that it should never be necessary to use this command unless the CRL file is removed by mistake or is corrupted and needs to be regenerated from scratch.

Server Keysize

The Local CA server keypair size can be configured independently of the user-issued certificate keypair size. The **keysize server** command is used to configure the size of the Local CAs own keypair. The **keysize** command specifies the size of the public and private keys generated at user-certificate enrollment. The **keysize server** command is illustrated in the following example:

```
hostname(config)# crypto ca server
hostname(config-ca-server)# keysize server 2048
hostname(config-ca-server)#
```

For both the keysize command and the **keysize server** command, key-pair size options are 512, 768, 1024, 2048 bits, and both commands have default values of 1024 bits.

Note The Local CA keysize cannot be changed once the Local CA is enabled without deleting the Local CA and reconfiguring a new Local CA. This would invalidate all issued certificates.

Defining Storage for Local CA Files

The security appliance accesses and implements user information, issued certificates, revocation lists, and so forth using a Local CA database. That database resides in local flash memory by default or can be configured to be on an off-box file system that is mounted and accessible to the security appliance.

Default Flash Memory Data Storage

By default, the Local CA server database is stored in flash memory, a nonvolatile storage space that stores the configuration and database files when the security appliance is powered down.

There are no limits on the number of users that can be in the Local CA user database; however, if flash memory storage issues arise, syslogs are generated to alert the administrator to take action, and the Local CA could be disabled until the storage problems are solved. Flash memory can store a database with 3500 users or less, but a database of more than 3500 users requires off-box storage.

Setting up External Local CA File Storage

Storage for Local CA files on a server external to the security appliance requires an already mounted file system of file type CIFS or FTP that is username- and password-protected to secure the stored information. With the file system mounted, you then can establish a path to the server and specify the file or folder name for the Local CA to use for file storage and retrieval.

Configure the file system path with the **database path** command. To return Local CA file storage to the security appliance flash memory, use the **no database path** command.

To specify external off-box storage for the Local CA, perform the following steps:

Step 1 Enter the **mount** command with a file system label and type in global configuration mode. This lets the security appliance access the configuration mode for the specific file system type. An example that mounts a CIFS file system follows:

```
hostname(config)# mount mydata type cifs
hostname(config-mount-cifs)# mount mydata type cifs
server 99.1.1.99 share myshare
domain frqa.ASC.com
username user6
password ********
status enable
hostname(config-mount-cifs)#
```

Step 2 Use the **database path** command to specify the location of mydata, the pre-mounted CIFS file system to be used for the Local CA server database.

```
hostname(config)# crypto ca server
hostname(config-ca-server)# database path mydata:newuser
hostname(config-ca-server)#
```

Note Only the user who mounts a file system can un-mount it with the **no mount** command.

CRL Storage

The Certificate Revocation List (CRL) exists for other devices to validate the revocation of certificates issued by the Local CA. In addition, the Local CA tracks all issued certificates and status within its own certificate database. Revocation checking is done when a validating party needs to validate a user certificate by retrieving the revocation status from an external server, which might be the CA that issued the certificate or a server designated by the CA.

If you do not configure a specific location for the CDP, the default location URL is http://*hostname.domain*/+CSCOCA+/asa_ca.crl. To establish a specific location for the Local CA's automatically generated CRL, use the **cdp-url** command to specify the certificate revocation list distribution point (CDP) to be included in all issued certificates. An example follows:

```
hostname(config)# crypto ca server
hostname(config-ca-server)# cdp-url http://99.1.1.99/pathname/myca.crl
```

The Local CA updates and reissues the CRL every time a user certificate is revoked or unrevoked. If there are no revocation changes, the CRL is reissued once every CRL lifetime, the period of time you specify with the **lifetime** command during Local CA configuration. An example follows:

If you do not specify a CRL lifetime, the default time period is six hours.

```
hostname(config)# crypto ca server
hostname (config-ca-server)# lifetime crl 72
```

If the **cdp-url** command is set to serve the CRL directly from the Local CA security appliance, use the **publish-crl** CLI command to open a port on an interface to make the CRL accessible from that interface. The **publish-crl** command is detailed in the following section.

CRL Downloading

To make the CRL available for HTTP download on a given interface or port, use the **publish-crl** command in config-ca-server mode. The specified interface and port are used to listen for incoming requests for the CRL. Interface options are:

inside	name of interface GigabitEthernet0/1
management	name of interface Management0/0
outside	name of interface GigabitEthernet0/0

The optional port option can be any port number in a range of 1-65535, and TCP port 80 is the HTTP default port number. For example, to specify port 70 for outside access to the CRL, use the following command:

```
hostname (config)# crypto ca server
hostname (config-ca-server)#publish-crl outside 70
hostname (config-ca-server)#
```

The CDP URL can be configured to utilize the IP address of an interface, and the path of the CDP URL and the file name can be configured also. For example, the CDP URL could be configured to be:
`http://10.10.10.100/user8/my_crl_file`

In this case only the interface with that IP address configured listens for CRL requests, and when a request comes in, the security appliance matches the path /user8/my_crl_file to the configured CDP URL. When the path matches, the security appliance returns the CRL file stored in storage. Note that the protocol must be http, so the prefix is http://.

Note If you do not specify a **publish-crl** command, the CRL is not accessible from the CDP location because the **publish-crl** command is required in order to open an interface for downloading the CRL file.

Enrolling Local CA Users

Each user who wishes to be enrolled as a Local CA user must be added to the Local CA server user database. User enrollment is initiated by the Local CA administrator who adds new users to the database with the **crypto ca server user-db add** command.

Next, the administrator issues a **crypto ca server user-db allow...** command, and, if email-OTP is specified, the Local CA Server e-mails a one-time-password and username to the new user to enable enrollment. The e-mail, an automatically generated message, contains the enrollment URL of the security appliance. Figure 41-2 shows a sample e-mail to a new user.

```
Date: 12/22/06
To: wuser6@wuser.com
From: Wuseradmin
Subject: Certificate Enrollment Invitation

You have been granted access to enroll for a certificate.

The credentials below can be used to obtain your certificate.
Username: wuser6@wuser.com
One-time Password: C93BBB733CD80C74
Enrollment is allowed until: 15:54:31 UTC Thu Dec 27 2006

NOTE: The one-time password is also used as the passphrase to unlock the certificate
file.

Please visit the following site to obtain your certificate:
https://wu5520-FO.frdevtestad.local/+CSCOCA+/enroll.html
You may be asked to verify the fingerprint/thumbprint of the CA certificate
during installation of the certificates. The fingerprint/thumbprint should be:
MD5: 76DD1439 AC94FDBC 74A0A89F CB815ACC
SHA1: 58754FFD 9F19F9FD B13B4B02 15B3E4BE B70B5A83
```

Figure 41-2 Sample Local CA Enrollment E-mail

When a user enrolls successfully, a PKCS12 file is created, which contains a keypair and a certificate issued to the user, along with the Local CA certificate. The user must browse to the enrollment interface and enter a valid username and one-time password. Once the Local CA authenticates the user's credentials within the enrollment time frame, the user is permitted to download the newly generated certificate, which is included in a PKCS12 file.

The PKCS12 file contents are protected by a passphrase, the One-Time-Password (OTP). The OTP can be handled manually, or this file can be e-mailed to the user by the Local CA to download once the administrator allows enrollment.

The file is saved to storage temporarily as username.p12. This file contains the user certificate, the keypair, and the Local CA certificate. To install these certificates on the user's PC, the user is prompted for the passphrase (one-time password) for the file, the same one-time password used to authenticate the user to the Local CA.

With the file in storage, the user can return within the enrollment-retrieval time period to retrieve the file a second or subsequent times as needed. When the time period expires, the file is removed from storage automatically and is no longer available for downloading.

Setting Up Enrollment Parameters

For a secure enrollment process, the Local CA automatically generates one-time passwords (OTPs), which are e-mailed to enrolling users at the e-mail address the administrator configures. OTPs can be handled manually but are e-mailed if configured with an e-mail address when the user is added to the database. In order to complete enrollment and receive a certificate, the user must enter the OTP in the enrollment interlace along with a username in order to complete enrollment.

Each unique OTP has a configurable window of time in which it can be used to retrieve a certificate. If the OTP expiration period expires before the user retrieves the PKCS12 enrollment file that contains the user certificate, enrollment is not permitted. The **otp expiration** command defines the amount of time the OTP is valid for user enrollment.

The **enrollment-retrieval** command specifies the time in hours that an enrolled user can retrieve a certificate. An example of setting up enrollment parameters follows:

Step 1 Enter the **crypto ca server** command to access the Local CA Server Configuration mode. An example follows:

```
hostname(config)# crypto ca server
hostname (config-ca-server)#
```

Step 2 Specify the number of hours (24) that an issued One-Time Password (OTP) for the local Certificate Authority (CA) enrollment page is valid with the **otp expiration** command. This time period begins when the user is allowed to enrol. The default expiration time of 72 hours can be changed to 24 as follows:

```
hostname(config-ca-server)# otp expiration 24
hostname(config-ca-server)#
```

Note The user OTP for enrolling for a certificate with the enrollment interface page is also used as the password to unlock the PKCS12 file containing that user's issued certificate and keypair.

Step 3 Specify the number of hours an already-enrolled user can retrieve a PKCS12 enrollment file with the **enrollment-retrieval** command. This time period begins when the user is successfully enrolled. This command modifies the default 24-hours retrieval period to any value between one and 720 hours. Note that enrollment retrieval period is independent of the OTP expiration period. The following example sets the retrieval time to 120 hours (five days).

```
hostname(config)# crypto ca server
hostname(config-ca-server)# enrollment-retrieval 120
hostname(config-ca-server)#
```

After the enrollment-retrieval time expires, the user certificate and keypair are no longer available; the only way for the user to receive a certificate is for the administrator to reinitialize certificate enrollment by allowing the user to enroll again.

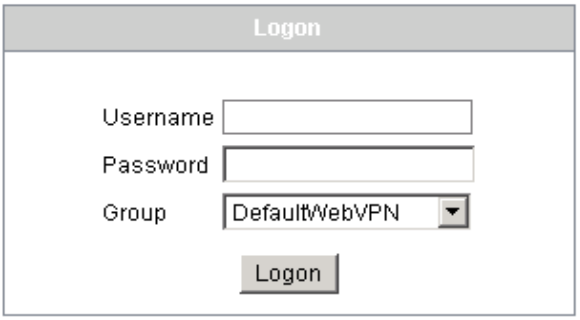
For the CLI commands that let you display and view the database entries, refer to the section Displaying Local CA Server Information further on in this chapter.

Enrollment Requirements

End-users enroll for a certificate by visiting the Local CA Enrollment Interface webpage and entering a username and one-time password. Enrolling as a user on the Local CA server initially requires valid user credentials, which typically are a username and a password.

When a user enrolls, the Local CA generates the user certificate and provides a link so the user can install the certificate on the client machine. The user's private keypair is generated by the Local CA and is issued to the user as part of the PKCS12 file. The PKCS12 file includes a keypair and the certificate issued to the user and the Local CA certificate.

The Local CA WebVPN login screen is provided in the following figure:



Starting and Stopping the Local CA Server

When you complete Local CA Server configuration, to activate it, use the **no shutdown** command. To disable enrollment and/or to modify the configuration, use the **shutdown** command

Enabling the Local CA Server

Initially, you need to specify a passphrase to create and protect the archive of the CA certificate and keypair that are generated. The passphrase unlocks the PKCS12 archive in case the CA certificate or keypair are lost.

Once you enable the Local CA server, with the **no shutdown** command, it generates the Local CA server certificate, keypair and necessary database files, and also archives the Local CA server certificate and keypair to storage in a PKCS12 file. After the initial startup, you can issue **no shutdown** and **shutdown** commands that enable and disable the Local CA without being prompted for the passphrase.

Note Once you enable the Local CA Server, be sure to save the configuration to ensure that the Local CA certificate and keypair are not lost after a reboot.

At initial startup, you are prompted for the passphrase in the CLI as illustrated in the example that follows. To enable the Local CA server on a security appliance, perform the following steps:

Step 1 Create a password (7-character min.) in order to encode and archive a PKCS12 file containing the Local CA certificate and keypair that is to be generated.

Step 2 Enter the following command to enable the Local CA server on the security appliance. The command requires an 8-65 alphanumeric character password:

```
hostname(config)# crypto ca server
hostname(config-ca-server)# no shutdown
hostname(config-ca-server)#

hostname(config-ca-server)# no shutdown
```

```
% Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key
% or type Return to exit

Password: caserver

Re-enter password: caserver

Keypair generation process begin. Please wait...

hostname(config-ca-server)#
```

Re-enabling the same Local CA Server with the **no shutdown** command and disabling it with the **shutdown** command do not require the passphrase.

Debugging the Local CA Server

To debug the newly configured Local CA Server, use the **debug crypto ca server** command in global configuration mode. This command displays debug messages when you configure and enable the Local CA server. By default, the **debug crypto ca server** command performs level 1 debug functions; levels 1-255 are available.

Note Debug commands might slow down traffic on busy networks. Levels 5 and higher are reserved for raw data dumps and should be avoided during normal debugging because of excessive debug output.

Disabling the Local CA Server

When you disable the Local CA server with the **shutdown** command, the configuration and all associated files remain in storage. Webpage enrollment is disabled, but you can change or reconfigure the Local CA Server during shutdown and then restart it with the **no shutdown** command.

To disable the Local CA server on a security appliance, perform the following:

```
asa1(config-ca-server)#
asa1(config-ca-server)# shutdown
INFO: Local CA Server has been shutdown.
asa1(config-ca-server)#
```

Managing the Local CA User Database

The Local CA server keeps track of user certificates, so the administrator can revoke or restore privileges as needed. This section describes how to add, allow for enrollment, remove, and manage users in the Local CA database with CLI commands. These operations are all initiated with the **crypto ca server user-db** *(function)* command in Privileged Exec mode. The functions are summarized inTable 41-2 Crypto CA Server User Commands and described in the following subsections.

Note that users must be added to the database with the **crypto ca server user-db add** command, but it is the **crypto ca server user-db allow** command that grants each user enrollment privileges.

Table 41-2 Crypto CA Server User Commands

Command	Description
crypto ca server user-db add	Adds a user to the Local CA server user database. If a DN string contains a comma, enclose the value string with double quotes (for example, O="Company, Inc.").
crypto ca server user-db allow	Permits a specific user or subset of users in the Local CA server database to enroll and generates OTPs for users.
crypto ca server user-db remove	Removes a user from the Local CA server user database by username.
crypto ca server user-db email-otp	E-mails the one-time password to a specific user or to a subset of users in the Local CA server database.
crypto ca server user-db show-otp	Displays the one-time password for a specific user or a subset of users in the Local CA server database.

Adding and Enrolling Users

Both the **crypto ca server user-db add** command and the **crypto ca server user-db allow** command are used to add and allow new Local CA users. To add a user who is eligible for enrollment to the Local CA database, perform the following steps:

Step 1 Add a new user with the following CLI commands:

```
hostname (config)#
hostname (config-ca-server)# crypto ca server user-db add username [dn dn] [email emailad-
dress]
hostname (config-ca-server)#
```

where the options are as follows:

- *username*—A string from 4-64 characters, the simple user name for the user being added. The username can be an e-mail address, which then is used to contact the user as necessary for enrollment invitations
- dn— distinguished name, a global, authoritative name of an entry in the OSI Directory (X.500), for example, cn=maryjane@ASC.com, cn=Engineer, o=ASC Systems, c=US. For details, see Customizing the Local CA Server
- e-mail-address—The e-mail address of the new user where OTPs and notices are to be sent.

Step 2 Provide user privileges to an added user with the following command:

```
hostname(config)#
hostname(config-ca-server)# crypto ca server user-db allow user6
hostname(config-ca-server)#
```

Step 3 Notify a user in the Local CA database to enroll and download a user certificate with the **crypto ca server user-db email-otp** command, which automatically e-mails the one-time password to that user.

```
hostname(config)#
hostname(config-ca-server)# crypto ca server user-db email-otp username
hostname(config-ca-server)#
```

If the user specifies the a-mail address in the **crypto ca server user-db add** command, it is to send the e-mail as part of the **crypto ca server user-db allow** command or after using the **crypto ca server user-db email-otp** command. When an administrator wants to be able to notify a user by means of e-mail, the e-mail address must be specified as the username or the e-mail field when adding the user.

Once a user is added with a valid e-mail address, the administrator has choice of **crypto ca server user-db allow** *username* **email-otp**, or **crypto ca server user-db allow** *username* and **crypto ca server user-db email-otp** *username*.

Alternatively, you could specify the email address in step 2, and omit the **crypto ca server user-db e-mail-otp** command. To view the one-time-password issued, use the **crypto ca server user-db show-otp** command. You can use a separate **show-otp** command in order to communicate the OTP to the user by other means

Once a user enrolls within the time limit with the correct OTP, the Local CA Server generates a keypair for the user and a user certificate based on the public key from the keypair generated and the subject-name DN specified with the DN field when the user is added or the subject-name-default setting if not specified. The enrollment time limit is set with the **otp-expiration** command, and the expiration date for the user certificate is specified during configuration with the **lifetime certificate** command.

Renewing Users

Renewing a user certificate is similar to the initial enrollment process. Each user certificate has an expiration date, and Local CA automatically reminds the user by e-mail to renew before the time period runs out. If a certificate expires, it becomes invalid. Renewal notices and the times they are e-mailed to users are variable and can be configured by the administrator during Local CA server configuration.

To specify the timing of renewal notices, use the **renewal-reminder** command to specify the number of days (1-90) prior to Local CA certificate expiration that an initial reminder to re-enroll is sent to certificate owners.

```
hostname(config)# crypto ca server
hostname(config-ca-server)# renewal-reminder 7

hostname(config-ca-server)#
```

There are three reminders in all, and an automatic e-mail goes out to the certificate owner for each of the three reminders, provided an e-mail address is specified in the user database. If no e-mail address exists for the user, a syslog message alerts you of the renewal requirement.

The security appliance automatically grants certificate renewal privileges to any user who holds a valid certificate that is about to expire provided the user still is in the user database. Therefore, if an administrator does not want to allow a user to renew automatically, the user must be removed from the database prior to the renewal time period.

Revoking Certificates and Removing or Restoring Users

Any time that user is to have a valid certificate revoked, use the **crypto ca server revoke** command to mark the certificate as revoked in the certificate database on the CA server and in the CRL, which is automatically reissued. To revoke a user certificate, enter the certificate serial number in hex format as shown in the following example, which revokes the certificate with the serial number 782ea09f:

```
hostname(config-ca-server)## crypto ca server revoke 782ea09f
Certificate with the serial number 0x782ea09f has been revoked. A new CRL has been issued.

hostname(config-ca-server)#
```

Note that the CRL is regenerated automatically after the specified certificate is revoked.

To restore a user and unrevoke a previously revoked certificate issued by the Local CA server, use the **crypto ca server unrevoke** command.

If you delete a user from the user database by username with the **crypto ca server user-db remove** command, you are prompted to permit revocation of any valid certificates issued to the user.

Revocation Checking

The Local CA maintains a current Certification Revocation List (CRL) with serial numbers of all revoked user certificates. This list is available to external devices and can be retrieved directly from the Local CA if it is configured as such with the **cdp-url** and the **publish-crl** CLI commands. When you revoke (or unrevoke) any current certificate, by certificate serial number, the CRL reflect these changes.

Displaying Local CA Server Information

There are various ways to display and print the Local CA server configuration and user information as described in the following subsections. The following table summarizes the Local CA Server CLI commands that display configuration and database information.

Command	Display
show crypto ca server	`Local CA configuration and status`
show crypto ca server cert-db	`User certificate(s)`
show crypto ca server certificate	`Local CA certificate`
show crypto ca server crl	`Certificate Revocation List`
show crypto ca server user-db	`Users and their status`
show crypto ca server user-db allowed	`Users eligible to enroll.`
show crypto ca server user-db enrolled	`Enrolled users with valid certificate`
show crypto ca server user-db expired	`Users with an expired certificate.`
show crypto ca server user-db on-hold	`Users without certificate not permitted to enroll`

Display Local CA Configuration

To display the characteristics of the configured Local CA, use the **show crypto ca server** command in Privileged EXEC mode. The following is a sample **show crypto ca server** display.

```
Certificate Server LOCAL-CA-SERVER:
   Status: enabled
   State: enabled
   Server's configuration is locked (enter “shutdown" to unlock it)
   Issuer name: CN=wz5520-1-16
   CA certificate fingerprint/thumbprint: (MD5)
      76dd1439 ac94fdbc 74a0a89f cb815acc
   CA certificate fingerprint/thumbprint: (SHA1)
      58754ffd 9f19f9fd b13b4b02 15b3e4be b70b5a83
   Last certificate issued serial number: 0x6
   CA certificate expiration timer: 14:25:11 UTC Jan 16 2008
   CRL NextUpdate timer: 16:09:55 UTC Jan 24 2007
   Current primary storage dir: flash:
```

Display Certificate Database

To display a list with all of the certificates issued by the Local CA, use the **show crypto ca server cert-db command** in Privileged EXEC mode. The following is a sample **show crypto ca server cert-db command** display showing just two of the user certificates in the database.

```
Username: emily1
Renewal allowed until: Not Allowed
Number of times user notified: 0
PKCS12 file stored until: 12:45:52 UTC Fri Jan 4 2008
Certificates Issued:
serial:   0x71
issued:   12:45:52 UTC Thu Jan 3 2008
expired:  12:17:37 UTC Sun Dec 31 2017
status:   Not Revoked
Username: fred1
Renewal allowed until: Not Allowed
Number of times user notified: 0
PKCS12 file stored until: 12:27:59 UTC Fri Jan 4 2008
Certificates Issued:
serial:   0x2
issued:   12:27:59 UTC Thu Jan 3 2008
expired:  12:17:37 UTC Sun Dec 31 2017
status:   Not Revoked
<--- More --->
```

Display the Local CA Certificate

To display the certificate of the Local CA on the console use the **show crypto ca server certificate** command in Privileged EXEC mode. The certificate displays in base 64 format and can be cut-and-pasted as an import into other devices that need the local CA certificate. A sample display follows:

The base64 encoded local CA certificate follows:

```
MIIXlwIBAzCCF1EGCSqGSIb3DQEHAaCCF0IEghc+MIIXOjCCFzYGCSqGSIb3DQEHBqCCFycwghcjAgEAMIIXHAYJKo
ZIhvcNAQcBMBsGCiqGSIb3DQEMAQMwDQQIjph4SxJoyTgCAQGAghbw3v4bFy+GGG2dJnB4OLphsUM+IG3SDOiDwZG9
n1SvtMieoxd7Hxknxbum06JDrujWKtHBIqkrm+td34qlNE1iGeP2YC94/NQ2z+4kS+uZzwcRhl1KEZTS1E4L0fSaC3
uMTxJq2NUHYWmoc8pi4CIeLj3h7VVMy6qbx2AC8I+q57+QG5vG5l5Hi5imwtYfaWwPEdPQxaWZPrzoG1J8BFqdPa1j
BGhAzzuSmElm3j/2dQ3Atro1G9nIsRHgV39fcBgwz4fEabHG7/Vanb+fj81d5nlOiJjDYYbP86tvbZ2yOVZR6aKFVI
0b2AfCr6PbwfC9U8Z/aF3BCyM2sN2xPJrXva94CaYrqyotZdAkSYA5KWScyEcgdqmuBeGDKOncTknfgy0XM+fG5rb3
qAXy1GkjyFI5Bm9Do6RUROoG1DSrQrKeq/hj….

END OF CERTIFICATE
```

Display the CRL

To display the Local CA CRL, use the **show crypto ca server crl** command as follows:

```
hostname(config)# show crypto ca server crl
Certificate Revocation List:
    Issuer: cn=xx5520-1-3-2007-1
    This Update: 13:32:53 UTC Jan 4 2008
    Next Update: 13:32:53 UTC Feb 3 2008
    Number of CRL entries: 2
    CRL size: 270 bytes
Revoked Certificates:
    Serial Number: 0x6f
    Revocation Date: 12:30:01 UTC Jan 4 2008
    Serial Number: 0x47
    Revocation Date: 13:32:48 UTC Jan 4 2008
hostname(config)#
```

Display the User Database

To display users in the CA server user database, use the **show crypto ca server user-db** command. This command can be used with qualifiers to reduce number of records displayed. Qualifiers are:

- `allowed show only users currently allowed to enroll.`
- `enrolled Show only users that are enrolled and hold a valid certificate`
- `expired  Show only users holding expired certificates.`
- `on-hold  List only users without a certificate and not currently allowed to enroll.`

The following example shows the resulting display (edited) for the entire database with no qualifiers.

```
hostname (config)# show crypto ca server user-db
username: wilma24
email:    wilma24@xxrown.com
```

```
dn:       CN=mycn,OU=Sales,O=ASC.com,L=Franklin,ST=Mass,C=US
allowed:  12:29:08 UTC Sun Jan 6 2008
notified: 1
.
.
.
username: wilma98
email:    wilma98@xxrown.com
dn:       CN=mycn,OU=Sales,O=ASC.com,L=Franklin,ST=Mass,C=US
allowed:  12:29:18 UTC Sun Jan 6 2008
notified: 1

username: wilma99
email:    wilma99@xxrown.com
dn:       CN=mycn,OU=Sales,O=ASC.com,L=Franklin,ST=Mass,C=US
allowed:  12:29:18 UTC Sun Jan 6 2008
notified: 1
hostname(config)#
```

The following example shows the display of the **show crypto ca server user-db** command when the **on-hold** qualifier is used yielding just one user on-hold:

```
hostname (config)# show crypto ca server user-db on-hold
username: wilma101
email:    <None>
dn:       <None>
allowed:  <not allowed>
notified: 0
hostname (config)#
```

Local CA Server Maintenance and Backup Procedures

The stored Local CA Server configuration, users, issued certificates, CRL, etc. reside in the database in flash memory, or in file-system storage, depending on how you configure storage. The following subsections describe database maintenance procedures.

Maintaining the Local CA User Database

Each time the security appliance configuration is saved, all user information in the Local CA Server database is saved automatically (with the **write memory** command) to the file specified by the **database path** command when you set up file storage external to the security appliance. For example, if you set up file storage using the following command:

```
hostname(config)# crypto ca server
hostname(config-ca-server)# database path mydata:newuser
hostname(config-ca-server)#
```

User database information is saved from the security appliance to *mydata /newuser* every time you save the security appliance configuration.

Note For flash memory database storage, the user information is saved automatically to the default location for the start-up configuration.

Maintaining the Local CA Certificate Database

The certificate database file, LOCAL-CA-SERVER.cdb, is to be saved anytime there is a change in the database.

- LOCAL-CA-SERVER.p12 is the archive of the Local CA certificate and keypair generated when the Local CA server is initially enabled with the **no shutdown** command.
- LOCAL-CA-SERVER.crl file is the actual CRL.
- LOCAL-CA-SERVER.ser file is used to keep track of the issued certificate serial numbers

The Local CA files can be seen on the flash memory or in external storage as follows:

```
hostname(config-ca-server)# dir LOCAL* //

Directory of disk0:/LOCAL*

75    -rwx  32        13:07:49 Jan 20 2007  LOCAL-CA-SERVER.ser
77    -rwx  229       13:07:49 Jan 20 2007  LOCAL-CA-SERVER.cdb
69    -rwx  0         01:09:28 Jan 20 2007  LOCAL-CA-SERVER.udb
81    -rwx  232       19:09:10 Jan 20 2007  LOCAL-CA-SERVER.crl
72    -rwx  1603      01:09:28 Jan 20 2007  LOCAL-CA-SERVER.p12

127119360 bytes total (79693824 bytes free)
hostname (config-ca-server)#
```

Local CA Certificate Rollover

Thirty days prior to the expiration of the Local CA certificate, a rollover replacement certificate is generated, and a syslog message informs the administrator that it is time for Local CA rollover. The new Local CA certificate must be imported onto all necessary devices prior to the expiration of the current certificate. If the administrator does not respond by installing the rollover certificate as the new Local CA certificate, validations can begin to fail.

The Local CA certificate rolls over automatically upon expiration using the same keypair. The rollover certificate is available for export in base64 format and can be displayed using the **crypto ca server certificate** command, which displays both the current and the rollover certificates. This command shows information about the rollover certificate when available, including the thumbprint of the rollover certificate for verification of the new certificate during import on other devices.

Archiving the Local CA Server Certificate and Keypair

For backup purposes, you can use FTP or TFTP to copy the Local CA Server certificate and keypair and all files from the security appliance. An example follows:

```
hostname#
hostname# copy LOCAL-CA-SERVER_0001.p12 tftp://90.1.1.22/user6/
```

Note Back up all Local CA files as often as possible.

Deleting the Local CA Server

Note Deleting the Local CA Server removes the configuration from the security appliance. Once deleted, the configuration is unrecoverable.

To delete the existing Local CA server, whether it is enabled or disabled, you must issue a **no crypto ca server** command or a **clear config crypto ca server** command in Global Configuration mode, and then delete the associated database and configuration files (all files with the wildcard name, LOCAL-CA-SERVER.*).



PART 4

System Administration



CHAPTER 42

Managing System Access

This chapter describes how to access the security appliance for system management through Telnet, SSH, and HTTPS (using ASDM). It also describes how to authenticate and authorize users and how to create login banners.

This chapter includes the following sections:

- Allowing Telnet Access, page 42-1
- Allowing SSH Access, page 42-2
- Allowing HTTPS Access for ASDM, page 42-3
- Managing the Security Appliance on a Different Interface from the VPN Tunnel Termination Interface, page 42-5
- Configuring AAA for System Administrators, page 42-5
- Configuring a Login Banner, page 42-20

Note To access the security appliance interface for management access, you do not also need an access list allowing the host IP address. You only need to configure management access according to the sections in this chapter.

Allowing Telnet Access

The security appliance allows Telnet connections to the security appliance for management purposes. You cannot use Telnet to the lowest security interface unless you use Telnet inside an IPSec tunnel.

The security appliance allows a maximum of 5 concurrent Telnet connections per context, if available, with a maximum of 100 connections divided between all contexts.

To configure Telnet access to the security appliance, follow these steps:

Step 1 To identify the IP addresses from which the security appliance accepts connections, enter the following command for each address or subnet:

```
hostname(config)# telnet source_IP_address mask source_interface
```

If there is only one interface, you can configure Telnet to access that interface as long as the interface has a security level of 100.

Step 2 (Optional) To set the duration for how long a Telnet session can be idle before the security appliance disconnects the session, enter the following command:

```
hostname(config)# telnet timeout minutes
```

Set the timeout from 1 to 1440 minutes. The default is 5 minutes. The default duration is too short in most cases and should be increased until all pre-production testing and troubleshooting has been completed.

For example, to let a host on the inside interface with an address of 192.168.1.2 access the security appliance, enter the following command:

```
hostname(config)# telnet 192.168.1.2 255.255.255.255 inside
hostname(config)# telnet timeout 30
```

To allow all users on the 192.168.3.0 network to access the security appliance on the inside interface, enter the following command:

```
hostname(config)# telnet 192.168.3.0 255.255.255.0 inside
```

Allowing SSH Access

The security appliance allows SSH connections to the security appliance for management purposes. The security appliance allows a maximum of 5 concurrent SSH connections per context, if available, with a maximum of 100 connections divided between all contexts.

SSH is an application running on top of a reliable transport layer, such as TCP/IP, that provides strong authentication and encryption capabilities. The security appliance supports the SSH remote shell functionality provided in SSH Versions 1 and 2 and supports DES and 3DES ciphers.

Note XML management over SSL and SSH are not supported.

This section includes the following topics:

- Configuring SSH Access, page 42-2
- Using an SSH Client, page 42-3

Configuring SSH Access

To configure SSH access to the security appliance, follow these steps:

Step 1 To generate an RSA key pair, which is required for SSH, enter the following command:

```
hostname(config)# crypto key generate rsa modulus modulus_size
```

The modulus (in bits) is 512, 768, 1024, or 2048. The larger the key modulus size you specify, the longer it takes to generate an RSA. We recommend a value of 1024.

Step 2 To save the RSA keys to persistent Flash memory, enter the following command:

```
hostname(config)# write mem
```

Step 3 To identify the IP addresses from which the security appliance accepts connections, enter the following command for each address or subnet:

```
hostname(config)# ssh source_IP_address mask source_interface
```

The security appliance accepts SSH connections from all interfaces, including the one with the lowest security level.

Step 4 (Optional) To set the duration for how long an SSH session can be idle before the security appliance disconnects the session, enter the following command:

```
hostname(config)# ssh timeout minutes
```

Set the timeout from 1 to 60 minutes. The default is 5 minutes. The default duration is too short in most cases and should be increased until all pre-production testing and troubleshooting has been completed.

For example, to generate RSA keys and let a host on the inside interface with an address of 192.168.1.2 access the security appliance, enter the following command:

```
hostname(config)# crypto key generate rsa modulus 1024
hostname(config)# write mem
hostname(config)# ssh 192.168.1.2 255.255.255.255 inside
hostname(config)# ssh 192.168.1.2 255.255.255.255 inside
hostname(config)# ssh timeout 30
```

To allow all users on the 192.168.3.0 network to access the security appliance on the inside interface, the following command:

```
hostname(config)# ssh 192.168.3.0 255.255.255.0 inside
```

By default SSH allows both version one and version two. To specify the version number enter the following command:

hostname(config)# **ssh version** *version_number*

The *version_number* can be *1* or *2*.

Using an SSH Client

To gain access to the security appliance console using SSH, at the SSH client enter the username **pix** and enter the login password set by the **password** command (see the "Changing the Login Password" section on page 9-1).

When starting an SSH session, a dot (.) displays on the security appliance console before the SSH user authentication prompt appears, as follows:

```
hostname(config)# .
```

The display of the dot does not affect the functionality of SSH. The dot appears at the console when generating a server key or decrypting a message using private keys during SSH key exchange before user authentication occurs. These tasks can take up to two minutes or longer. The dot is a progress indicator that verifies that the security appliance is busy and has not hung.

Allowing HTTPS Access for ASDM

To use ASDM, you need to enable the HTTPS server, and allow HTTPS connections to the security appliance. All of these tasks are completed if you use the **setup** command. This section describes how to manually configure ASDM access and how to login to ASDM.

The security appliance allows a maximum of 5 concurrent ASDM instances per context, if available, with a maximum of 32 ASDM instances between all contexts.

This section includes the following topics:

- Enabling HTTPS Access, page 42-4
- Accessing ASDM from Your PC, page 42-4

Enabling HTTPS Access

To configure ASDM access, follow these steps:

Step 1 To identify the IP addresses from which the security appliance accepts HTTPS connections, enter the following command for each address or subnet:

```
hostname(config)# http source_IP_address mask source_interface
```

Step 2 To enable the HTTPS server, enter the following command:

```
hostname(config)# http server enable [port]
```

By default, the *port* is 443. If you change the port number, be sure to include the new port in the ASDM access URL. For example, if you change it to port 444, enter:

```
https://10.1.1.1:444
```

Step 3 To specify the location of the ASDM image, enter the following command:

```
hostname(config)# asdm image disk0:/asdmfile
```

For example, to enable the HTTPS server and let a host on the inside interface with an address of 192.168.1.2 access ASDM, enter the following commands:

```
hostname(config)# crypto key generate rsa modulus 1024
hostname(config)# write mem
hostname(config)# http server enable
hostname(config)# http 192.168.1.2 255.255.255.255 inside
```

To allow all users on the 192.168.3.0 network to access ASDM on the inside interface, enter the following command:

```
hostname(config)# http 192.168.3.0 255.255.255.0 inside
```

Accessing ASDM from Your PC

From a supported web browser on the security appliance network, enter the following URL:

```
https://interface_ip_address[:port]
```

In transparent firewall mode, enter the management IP address.

Managing the Security Appliance on a Different Interface from the VPN Tunnel Termination Interface

If your IPSec VPN tunnel terminates on one interface, but you want to manage the security appliance by accessing a different interface, then enter the following command:

```
hostname(config)# management access management_interface
```

where *management_interface* specifies the name of the management interface you want to access when entering the security appliance from another interface.

For example, if you enter the security appliance from the outside interface, this command lets you connect to the inside interface using Telnet; or you can ping the inside interface when entering from the outside interface.

You can define only one management-access interface.

Configuring AAA for System Administrators

This section describes how to enable authentication and command authorization for system administrators. Before you configure AAA for system administrators, first configure the local database or AAA server according to Chapter 14, "AAA Server and Local Database Support."

This section includes the following topics:

- Configuring Authentication for CLI and ASDM Access, page 42-5
- Configuring Authentication To Access Privileged EXEC Mode (the enable Command), page 42-6
- Limiting User CLI and ASDM Access with Management Authorization, page 42-7
- Configuring Command Authorization, page 42-8
- Configuring Command Accounting, page 42-18
- Viewing the Current Logged-In User, page 42-18
- Recovering from a Lockout, page 42-19

Configuring Authentication for CLI and ASDM Access

If you enable CLI authentication, the security appliance prompts you for your username and password to log in. After you enter your information, you have access to user EXEC mode.

To enter privileged EXEC mode, enter the **enable** command or the **login** command (if you are using the local database only).

If you configure **enable** authentication (see the "Configuring Authentication for the enable Command" section on page 42-6), the security appliance prompts you for your username and password. If you do not configure **enable** authentication, enter the system enable password when you enter the **enable** command (set by the **enable password** command). However, if you do not use **enable** authentication, after you enter the **enable** command, you are no longer logged in as a particular user. To maintain your username, use **enable** authentication.

For authentication using the local database, you can use the **login** command, which maintains the username but requires no configuration to turn on authentication.

Note Before the security appliance can authenticate a Telnet, SSH, or HTTP user, you must first configure access to the security appliance using the **telnet**, **ssh**, and **http** commands. These commands identify the IP addresses that are allowed to communicate with the security appliance.

To authenticate users who access the CLI, enter the following command:

```
hostname(config)# aaa authentication {telnet | ssh | http | serial} console {LOCAL | server_group [LOCAL]}
```

The **http** keyword authenticates the ASDM client that accesses the security appliance using HTTPS. You only need to configure HTTP authentication if you want to use a AAA server. By default, ASDM uses the local database for authentication even if you do not configure this command. HTTP management authentication does not support the SDI protocol for a AAA server group.

If you use a AAA server group for authentication, you can configure the security appliance to use the local database as a fallback method if the AAA server is unavailable. Specify the server group name followed by **LOCAL** (**LOCAL** is case sensitive). We recommend that you use the same username and password in the local database as the AAA server because the security appliance prompt does not give any indication which method is being used.

You can alternatively use the local database as your main method of authentication (with no fallback) by entering **LOCAL** alone.

Configuring Authentication To Access Privileged EXEC Mode (the enable Command)

You can configure the security appliance to authenticate users with a AAA server or the local database when they enter the **enable** command. Alternatively, users are automatically authenticated with the local database when they enter the **login** command, which also accesses privileged EXEC mode depending on the user level in the local database.

This section includes the following topics:

- Configuring Authentication for the enable Command, page 42-6
- Authenticating Users Using the Login Command, page 42-7

Configuring Authentication for the enable Command

You can configure the security appliance to authenticate users when they enter the **enable** command. If you do not authenticate the **enable** command, when you enter **enable**, the security appliance prompts for the system enable password (set by the **enable password** command), and you are no longer logged in as a particular user. Applying authentication to the **enable** command maintains the username. This feature is particularly useful when you perform command authorization, where usernames are important to determine the commands a user can enter.

To authenticate users who enter the **enable** command, enter the following command:

```
hostname(config)# aaa authentication enable console {LOCAL | server_group [LOCAL]}
```

The user is prompted for the username and password.

If you use a AAA server group for authentication, you can configure the security appliance to use the local database as a fallback method if the AAA server is unavailable. Specify the server group name followed by **LOCAL** (**LOCAL** is case sensitive). We recommend that you use the same username and password in the local database as the AAA server because the security appliance prompt does not give any indication which method is being used.

You can alternatively use the local database as your main method of authentication (with no fallback) by entering **LOCAL** alone.

Authenticating Users Using the Login Command

From user EXEC mode, you can log in as any username in the local database using the **login** command.

This feature allows users to log in with their own username and password to access privileged EXEC mode, so you do not have to give out the system enable password to everyone. To allow users to access privileged EXEC mode (and all commands) when they log in, set the user privilege level to 2 (the default) through 15. If you configure local command authorization, then the user can only enter commands assigned to that privilege level or lower. See the "Configuring Local Command Authorization" section on page 42-11 for more information.

Caution If you add users to the local database who can gain access to the CLI and whom you do not want to enter privileged EXEC mode, you should configure command authorization. Without command authorization, users can access privileged EXEC mode (and all commands) at the CLI using their own password if their privilege level is 2 or greater (2 is the default). Alternatively, you can use a AAA server for authentication, or you can set all local users to level 1 so you can control who can use the system enable password to access privileged EXEC mode.

To log in as a user from the local database, enter the following command:

```
hostname> login
```

The security appliance prompts for your username and password. After you enter your password, the security appliance places you in the privilege level that the local database specifies.

Limiting User CLI and ASDM Access with Management Authorization

If you configure CLI or **enable** authentication, you can limit a local user, RADIUS, TACACS+, or LDAP user (if you map LDAP attributes to RADIUS attributes) from accessing the CLI, ASDM, or the **enable** command.

Note Serial access is not included in management authorization, so if you configure **aaa authentication serial console**, then any user who authenticates can access the console port.

To configure management authorization, perform the following steps:

Step 1 To enable management authorization, enter the following command:

```
hostname(config)# aaa authorization exec authentication-server
```

This command also enables support of administrative user privilege levels from RADIUS, which can be used in conjunction with local command privilege levels for command authorization. See the "Configuring Local Command Authorization" section on page 42-11 for more information.

Step 2 To configure the user for management authorization, see the following requirements for each AAA server type or local user:

- RADIUS or LDAP (mapped) users—Use the IETF RADIUS numeric Service-Type attribute which maps to one of the following values. (To map LDAP attributes, see the "LDAP Attribute Mapping" section on page 14-15.)
  - Service-Type 6 (Administrative)—Allows full access to any services specified by the **aaa authentication console** commands.
  - Service-Type 7 (NAS prompt)—Allows access to the CLI when you configure the **aaa authentication {telnet | ssh} console** command, but denies ASDM configuration access if you configure the **aaa authentication http console** command. ASDM monitoring access is allowed. If you configure **enable** authentication with the **aaa authentication enable console** command, the user cannot access privileged EXEC mode using the **enable** command.
  - Service-Type 5 (Outbound)—Denies management access. The user cannot use any services specified by the **aaa authentication console** commands (excluding the **serial** keyword; serial access is allowed). Remote access (IPSec and SSL) users can still authenticate and terminate their remote access sessions.
- TACACS+ users—Authorization is requested with the "service=shell" and the server responds with PASS or FAIL.
  - PASS, privilege level 1—Allows full access to any services specified by the **aaa authentication console** commands.
  - PASS, privilege level 2 and higher—Allows access to the CLI when you configure the **aaa authentication {telnet | ssh} console** command, but denies ASDM configuration access if you configure the **aaa authentication http console** command. ASDM monitoring access is allowed. If you configure **enable** authentication with the **aaa authentication enable console** command, the user cannot access privileged EXEC mode using the **enable** command.
  - FAIL—Denies management access. The user cannot use any services specified by the **aaa authentication console** commands (excluding the **serial** keyword; serial access is allowed).
- Local users—Set the **service-type** command. See the "Configuring the Local Database" section on page 14-7. By default, the **service-type** is **admin**, which allows full access to any services specified by the **aaa authentication console** commands.

Configuring Command Authorization

If you want to control the access to commands, the security appliance lets you configure command authorization, where you can determine which commands that are available to a user. By default when you log in, you can access user EXEC mode, which offers only minimal commands. When you enter the **enable** command (or the **login** command when you use the local database), you can access privileged EXEC mode and advanced commands, including configuration commands.

This section includes the following topics:

- Command Authorization Overview, page 42-9
- Configuring Local Command Authorization, page 42-11

- Configuring TACACS+ Command Authorization, page 42-14

Command Authorization Overview

This section describes command authorization, and includes the following topics:

- Supported Command Authorization Methods, page 42-9
- About Preserving User Credentials, page 42-9
- Security Contexts and Command Authorization, page 42-10

Supported Command Authorization Methods

You can use one of two command authorization methods:

- Local privilege levels—Configure the command privilege levels on the security appliance. When a local, RADIUS, or LDAP (if you map LDAP attributes to RADIUS attributes) user authenticates for CLI access, the security appliance places that user in the privilege level that is defined by the local database, RADIUS, or LDAP server. The user can access commands at the user's privilege level and below. Note that all users access user EXEC mode when they first log in (commands at level 0 or 1). The user needs to authenticate again with the **enable** command to access privileged EXEC mode (commands at level 2 or higher), or they can log in with the **login** command (local database only).

Note You can use local command authorization without any users in the local database and without CLI or **enable** authentication. Instead, when you enter the **enable** command, you enter the system enable password, and the security appliance places you in level 15. You can then create enable passwords for every level, so that when you enter **enable** *n* (2 to 15), the security appliance places you in level *n*. These levels are not used unless you turn on local command authorization (see "Configuring Local Command Authorization" below). (See the *Cisco Security Appliance Command Reference* for more information about **enable**.)

- TACACS+ server privilege levels—On the TACACS+ server, configure the commands that a user or group can use after they authenticate for CLI access. Every command that a user enters at the CLI is checked with the TACACS+ server.

About Preserving User Credentials

When a user logs into the security appliance, they are required to provide a username and password for authentication. The security appliance retains these session credentials in case further authentication is needed later in the session.

When the following configurations are in place, a user needs only to authenticate with the local server upon login. Subsequent serial authorization uses the saved credentials. The user is also prompted for the privilege level 15 password. When exiting privileged mode, the user is authenticated again. User credentials are not retained in privileged mode.

- Local server is configured to authenticate user access.
- Privilege level 15 command access is configured to require a password.
- User's account is configured for serial only authorization (no access to console or ASDM).
- User's account is configured for privilege level 15 command access.

The following table shows how credentials are used in this case by the security appliance.

Credentials required	Username and Password Authentication	Serial Authorization	Privileged Mode Command Authorization	Privileged Mode Exit Authorization
Username	Yes	No	No	Yes
Password	Yes	No	No	Yes
Privileged Mode Password	No	No	Yes	No

Security Contexts and Command Authorization

The following are important points to consider when implementing command authorization with multiple security contexts:

- AAA settings are discrete per context, not shared between contexts.

  When configuring command authorization, you must configure each security context separately. This provides you the opportunity to enforce different command authorizations for different security contexts.

  When switching between security contexts, administrators should be aware that the commands permitted for the username specified when they login may be different in the new context session or that command authorization may not be configured at all in the new context. Failure to understand that command authorizations may differ between security contexts could confuse an administrator. This behavior is further complicated by the next point.

- New context sessions started with the **changeto** command always use the default “enable_15” username as the administrator identity, regardless of what username was used in the previous context session. This behavior can lead to confusion if command authorization is not configured for the enable_15 user or if authorizations are different for the enable_15 user than for the user in the previous context session.

  This behavior also affects command accounting, which is useful only if you can accurately associate each command that is issued with a particular administrator. Because all administrators with permission to use the **changeto** command can use the enable_15 username in other contexts, command accounting records may not readily identify who was logged in as the enable_15 username. If you use different accounting servers for each context, tracking who was using the enable_15 username requires correlating the data from several servers.

  When configuring command authorization, consider the following:

  – An administrator with permission to use the **changeto** command effectively has permission to use all commands permitted to the enable_15 user in each of the other contexts.

  – If you intend to authorize commands differently per context, ensure that in each context the enable_15 username is denied use of commands that are also denied to administrators who are permitted use of the **changeto** command.

  When switching between security contexts, administrators can exit privileged EXEC mode and enter the **enable** command again to use the username they need.

Note The system execution space does not support AAA commands; therefore, command authorization is not available in the system execution space.

Configuring Local Command Authorization

Local command authorization lets you assign commands to one of 16 privilege levels (0 to 15). By default, each command is assigned either to privilege level 0 or 15. You can define each user to be at a specific privilege level, and each user can enter any command at their privilege level or below. The security appliance supports user privilege levels defined in the local database, a RADIUS server, or an LDAP server (if you map LDAP attributes to RADIUS attributes. See the "LDAP Attribute Mapping" section on page 14-15.)

This section includes the following topics:

- Local Command Authorization Prerequisites, page 42-11
- Default Command Privilege Levels, page 42-11
- Assigning Privilege Levels to Commands and Enabling Authorization, page 42-12
- Viewing Command Privilege Levels, page 42-13

Local Command Authorization Prerequisites

Complete the following tasks as part of your command authorization configuration:

- Configure **enable** authentication. (See the "Configuring Authentication To Access Privileged EXEC Mode (the enable Command)" section on page 42-6.)

  enable authentication is essential to maintain the username after the user accesses the **enable** command.

  Alternatively, you can use the **login** command (which is the same as the **enable** command with authentication; for the local database only), which requires no configuration. We do not recommend this option because it is not as secure as **enable** authentication.

  You can also use CLI authentication, but it is not required.

- See the following prerequisites for each user type:
  - Local database users—Configure each user in the local database at a privilege level from 0 to 15.

    To configure the local database, see the "Configuring the Local Database" section on page 14-7.
  - RADIUS users—Configure the user with Cisco VSA CVPN3000-Privilege-Level with a value between 0 and 15.
  - LDAP users—Configure the user with a privilege level between 0 and 15, and then map the LDAP attribute to Cisco VAS CVPN3000-Privilege-Level according to the "LDAP Attribute Mapping" section on page 14-15.

Default Command Privilege Levels

By default, the following commands are assigned to privilege level 0. All other commands are at level 15.

- **show checksum**
- **show curpriv**
- **enable**
- **help**
- **show history**
- **login**

- **logout**
- **pager**
- **show pager**
- **clear pager**
- **quit**
- **show version**

If you move any configure mode commands to a lower level than 15, be sure to move the **configure** command to that level as well, otherwise, the user will not be able to enter configuration mode.

To view all privilege levels, see the "Viewing Command Privilege Levels" section on page 42-13.

Assigning Privilege Levels to Commands and Enabling Authorization

To assign a command to a new privilege level, and enable authorization, follow these steps:

Step 1 To assign a command to a privilege level, enter the following command:

```
hostname(config)# privilege [show | clear | cmd] level level [mode {enable | cmd}] command
command
```

Repeat this command for each command you want to reassign.

See the following information about the options in this command:

- **show | clear | cmd**—These optional keywords let you set the privilege only for the show, clear, or configure form of the command. The configure form of the command is typically the form that causes a configuration change, either as the unmodified command (without the **show** or **clear** prefix) or as the **no** form. If you do not use one of these keywords, all forms of the command are affected.
- **level** *level*—A level between 0 and 15.
- **mode** {**enable | configure**}—If a command can be entered in user EXEC/privileged EXEC mode as well as configuration mode, and the command performs different actions in each mode, you can set the privilege level for these modes separately:
  - **enable**—Specifies both user EXEC mode and privileged EXEC mode.
  - **configure**—Specifies configuration mode, accessed using the **configure terminal** command.
- **command** *command*—The command you are configuring. You can only configure the privilege level of the *main* command. For example, you can configure the level of all **aaa** commands, but not the level of the **aaa authentication** command and the **aaa authorization** command separately.

Step 2 To support administrative user privilege levels from RADIUS, enter the following command:

```
hostname(config)# aaa authorization exec authentication-server
```

Without this command, the security appliance only supports privilege levels for local database users and defaults all other types of users to level 15.

This command also enables management authorization for local, RADIUS, LDAP (mapped), and TACACS+ users. See the "Limiting User CLI and ASDM Access with Management Authorization" section on page 42-7 for more information.

Step 3 To enable the use of local command privilege levels, which can be checked against the privilege level of users in the local database, RADIUS server, or LDAP server (with mapped attributes), enter the following command:

```
hostname(config)# aaa authorization command LOCAL
```

When you set command privilege levels, command authorization does not take place unless you configure command authorization with this command.

For example, the **filter** command has the following forms:

- **filter** (represented by the **configure** option)
- **show running-config filter**
- **clear configure filter**

You can set the privilege level separately for each form, or set the same privilege level for all forms by omitting this option. For example, set each form separately as follows.

```
hostname(config)# privilege show level 5 command filter
hostname(config)# privilege clear level 10 command filter
hostname(config)# privilege cmd level 10 command filter
```

Alternatively, you can set all filter commands to the same level:

```
hostname(config)# privilege level 5 command filter
```

The **show privilege** command separates the forms in the display.

The following example shows the use of the **mode** keyword. The **enable** command must be entered from user EXEC mode, while the **enable password** command, which is accessible in configuration mode, requires the highest privilege level.

```
hostname(config)# privilege cmd level 0 mode enable command enable
hostname(config)# privilege cmd level 15 mode cmd command enable
hostname(config)# privilege show level 15 mode cmd command enable
```

This example shows an additional command, the **configure** command, that uses the **mode** keyword:

```
hostname(config)# privilege show level 5 mode cmd command configure
hostname(config)# privilege clear level 15 mode cmd command configure
hostname(config)# privilege cmd level 15 mode cmd command configure
hostname(config)# privilege cmd level 15 mode enable command configure
```

Note This last line is for the **configure terminal** command.

Viewing Command Privilege Levels

The following commands let you view privilege levels for commands.

- To show all commands, enter the following command:

  ```
  hostname(config)# show running-config all privilege all
  ```

- To show commands for a specific level, enter the following command:

  ```
  hostname(config)# show running-config privilege level level
  ```

  The *level* is an integer between 0 and 15.

- To show the level of a specific command, enter the following command:

  ```
  hostname(config)# show running-config privilege command command
  ```

For example, for the **show running-config all privilege all** command, the system displays the current assignment of each CLI command to a privilege level. The following is sample output from the command.

```
hostname(config)# show running-config all privilege all
privilege show level 15 command aaa
privilege clear level 15 command aaa
privilege configure level 15 command aaa
privilege show level 15 command aaa-server
privilege clear level 15 command aaa-server
privilege configure level 15 command aaa-server
privilege show level 15 command access-group
privilege clear level 15 command access-group
privilege configure level 15 command access-group
privilege show level 15 command access-list
privilege clear level 15 command access-list
privilege configure level 15 command access-list
privilege show level 15 command activation-key
privilege configure level 15 command activation-key
....
```

The following command displays the command assignments for privilege level 10:

```
hostname(config)# show running-config privilege level 10
privilege show level 10 command aaa
```

The following command displays the command assignment for the **access-list** command:

```
hostname(config)# show running-config privilege command access-list
privilege show level 15 command access-list
privilege clear level 15 command access-list
privilege configure level 15 command access-list
```

Configuring TACACS+ Command Authorization

If you enable TACACS+ command authorization, and a user enters a command at the CLI, the security appliance sends the command and username to the TACACS+ server to determine if the command is authorized.

When configuring command authorization with a TACACS+ server, do not save your configuration until you are sure it works the way you want. If you get locked out because of a mistake, you can usually recover access by restarting the security appliance. If you still get locked out, see the "Recovering from a Lockout" section on page 42-19.

Be sure that your TACACS+ system is completely stable and reliable. The necessary level of reliability typically requires that you have a fully redundant TACACS+ server system and fully redundant connectivity to the security appliance. For example, in your TACACS+ server pool, include one server connected to interface 1, and another to interface 2. You can also configure local command authorization as a fallback method if the TACACS+ server is unavailable. In this case, you need to configure local users and command privilege levels according to the "Configuring Command Authorization" section on page 42-8.

This section includes the following topics:

- TACACS+ Command Authorization Prerequisites, page 42-15
- Configuring Commands on the TACACS+ Server, page 42-15
- Enabling TACACS+ Command Authorization, page 42-17

TACACS+ Command Authorization Prerequisites

Complete the following tasks as part of your command authorization configuration:

- Configure CLI authentication (see the "Configuring Local Command Authorization" section on page 42-11).
- Configure **enable** authentication (see the "Configuring Authentication To Access Privileged EXEC Mode (the enable Command)" section on page 42-6).

Configuring Commands on the TACACS+ Server

You can configure commands on a Cisco Secure Access Control Server (ACS) TACACS+ server as a shared profile component, for a group, or for individual users. For third-party TACACS+ servers, see your server documentation for more information about command authorization support.

See the following guidelines for configuring commands in Cisco Secure ACS Version 3.1; many of these guidelines also apply to third-party servers:

- The security appliance sends the commands to be authorized as "shell" commands, so configure the commands on the TACACS+ server as shell commands.

Note Cisco Secure ACS might include a command type called "pix-shell." Do not use this type for security appliance command authorization.

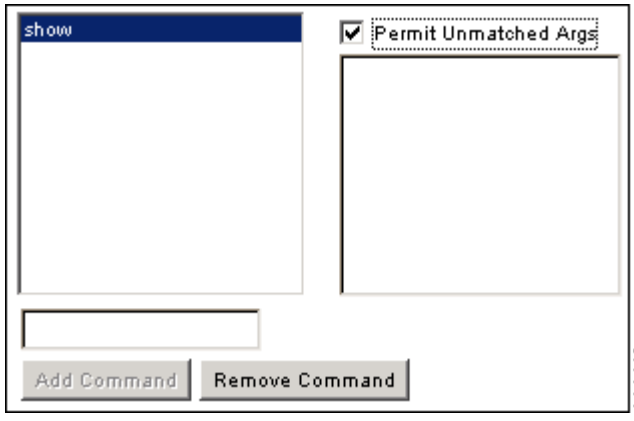
- The first word of the command is considered to be the main command. All additional words are considered to be arguments, which need to be preceded by **permit** or **deny**.

  For example, to allow the **show running-configuration aaa-server** command, add **show running-configuration** to the command box, and type **permit aaa-server** in the arguments box.

- You can permit all arguments of a command that you do not explicitly deny by selecting the **Permit Unmatched Args** check box.

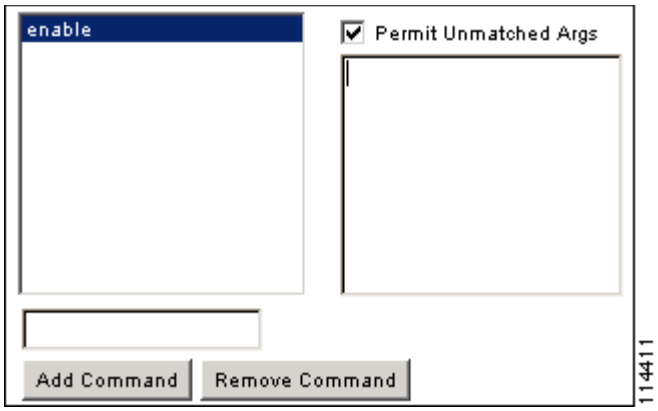
  For example, you can configure just the **show** command, and then all the **show** commands are allowed. We recommend using this method so that you do not have to anticipate every variant of a command, including abbreviations and **?**, which shows CLI usage (see Figure 42-1).

Figure 42-1 Permitting All Related Commands



- For commands that are a single word, you *must* permit unmatched arguments, even if there are no arguments for the command, for example **enable** or **help** (see Figure 42-2).

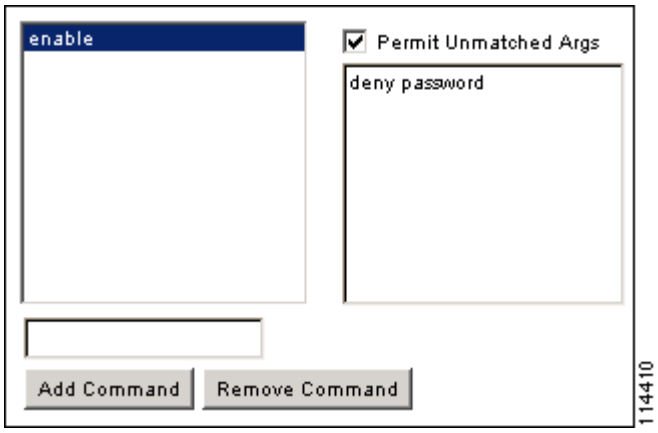
Figure 42-2 Permitting Single Word Commands



- To disallow some arguments, enter the arguments preceded by **deny**.

  For example, to allow **enable**, but not **enable password**, enter **enable** in the commands box, and **deny password** in the arguments box. Be sure to select the Permit Unmatched Args check box so that **enable** alone is still allowed (see Figure 42-3).

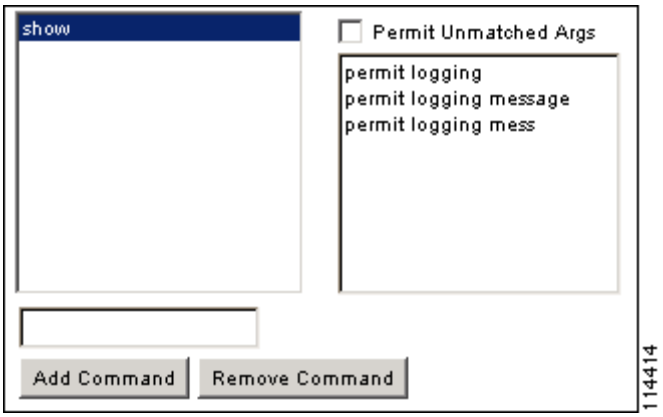
Figure 42-3 Disallowing Arguments



- When you abbreviate a command at the command line, the security appliance expands the prefix and main command to the full text, but it sends additional arguments to the TACACS+ server as you enter them.

  For example, if you enter **sh log**, then the security appliance sends the entire command to the TACACS+ server, **show logging**. However, if you enter **sh log mess**, then the security appliance sends **show logging mess** to the TACACS+ server, and not the expanded command **show logging message**. You can configure multiple spellings of the same argument to anticipate abbreviations (see Figure 42-4).

Figure 42-4 Specifying Abbreviations



- We recommend that you allow the following basic commands for all users:
  - **show checksum**
  - **show curpriv**
  - **enable**
  - **help**
  - **show history**
  - **login**
  - **logout**
  - **pager**
  - **show pager**
  - **clear pager**
  - **quit**
  - **show version**

Enabling TACACS+ Command Authorization

Before you enable TACACS+ command authorization, be sure that you are logged into the security appliance as a user that is defined on the TACACS+ server, and that you have the necessary command authorization to continue configuring the security appliance. For example, you should log in as an admin user with all commands authorized. Otherwise, you could become unintentionally locked out.

To perform command authorization using a TACACS+ server, enter the following command:

```
hostname(config)# aaa authorization command tacacs+_server_group [LOCAL]
```

You can configure the security appliance to use the local database as a fallback method if the TACACS+ server is unavailable. To enable fallback, specify the server group name followed by **LOCAL** (**LOCAL** is case sensitive). We recommend that you use the same username and password in the local database as the TACACS+ server because the security appliance prompt does not give any indication which method is being used. Be sure to configure users in the local database (see the "Configuring Command Authorization" section on page 42-8) and command privilege levels (see the "Configuring Local Command Authorization" section on page 42-11).

Configuring Command Accounting

You can send accounting messages to the TACACS+ accounting server when you enter any command other than **show** commands at the CLI. If you customize the command privilege level using the **privilege** command (see the "Assigning Privilege Levels to Commands and Enabling Authorization" section on page 42-12), you can limit which commands the security appliance accounts for by specifying a minimum privilege level. The security appliance does not account for commands that are below the minimum privilege level.

To enable command accounting, enter the following command:

```
hostname(config)# aaa accounting command [privilege level] server-tag
```

Where *level* is the minimum privilege level and *server-tag* is the name of the TACACS+ server group that to which the security appliance should send command accounting messages. The TACACS+ server group configuration must already exist. For information about configuring a AAA server group, see the "Identifying AAA Server Groups and Servers" section on page 14-9.

Viewing the Current Logged-In User

To view the current logged-in user, enter the following command:

```
hostname# show curpriv
```

See the following sample **show curpriv** command output. A description of each field follows.

```
hostname# show curpriv
Username : admin
Current privilege level : 15
Current Mode/s : P_PRIV
```

Table 42-1 describes the **show curpriv** command output.

Table 42-1 show curpriv Display Description

Field	Description
Username	Username. If you are logged in as the default user, the name is enable_1 (user EXEC) or enable_15 (privileged EXEC).
Current privilege level	Level from 0 to 15. Unless you configure local command authorization and assign commands to intermediate privilege levels, levels 0 and 15 are the only levels that are used.
Current Mode/s	Shows the access modes: • P_UNPR—User EXEC mode (levels 0 and 1) • P_PRIV—Privileged EXEC mode (levels 2 to 15) • P_CONF—Configuration mode

Recovering from a Lockout

In some circumstances, when you turn on command authorization or CLI authentication, you can be locked out of the security appliance CLI. You can usually recover access by restarting the security appliance. However, if you already saved your configuration, you might be locked out. Table 42-2 lists the common lockout conditions and how you might recover from them.

Table 42-2 CLI Authentication and Command Authorization Lockout Scenarios

Feature	Lockout Condition	Description	Workaround: Single Mode	Workaround: Multiple Mode
Local CLI authentication	No users in the local database	If you have no users in the local database, you cannot log in, and you cannot add any users.	Log in and reset the passwords and **aaa** commands.	Session into the security appliance from the switch. From the system execution space, you can change to the context and add a user.
TACACS+ command authorization TACACS+ CLI authentication RADIUS CLI authentication	Server down or unreachable and you do not have the fallback method configured	If the server is unreachable, then you cannot log in or enter any commands.	1. Log in and reset the passwords and AAA commands. 2. Configure the local database as a fallback method so you do not get locked out when the server is down.	1. If the server is unreachable because the network configuration is incorrect on the security appliance, session into the security appliance from the switch. From the system execution space, you can change to the context and reconfigure your network settings. 2. Configure the local database as a fallback method so you do not get locked out when the server is down.
TACACS+ command authorization	You are logged in as a user without enough privileges or as a user that does not exist	You enable command authorization, but then find that the user cannot enter any more commands.	Fix the TACACS+ server user account. If you do not have access to the TACACS+ server and you need to configure the security appliance immediately, then log into the maintenance partition and reset the passwords and **aaa** commands.	Session into the security appliance from the switch. From the system execution space, you can change to the context and complete the configuration changes. You can also disable command authorization until you fix the TACACS+ configuration.
Local command authorization	You are logged in as a user without enough privileges	You enable command authorization, but then find that the user cannot enter any more commands.	Log in and reset the passwords and **aaa** commands.	Session into the security appliance from the switch. From the system execution space, you can change to the context and change the user level.

Configuring a Login Banner

You can configure a message to display when a user connects to the security appliance, before a user logs in, or before a user enters privileged EXEC mode.

To configure a login banner, enter the following command in the system execution space or within a context:

```
hostname(config)# banner {exec | login | motd} text
```

Adds a banner to display at one of three times: when a user first connects (message-of-the-day (**motd**)), when a user logs in (**login**), and when a user accesses privileged EXEC mode (**exec**). When a user connects to the security appliance, the message-of-the-day banner appears first, followed by the login banner and prompts. After the user successfully logs in to the security appliance, the exec banner displays.

For the banner text, spaces are allowed but tabs cannot be entered using the CLI. You can dynamically add the hostname or domain name of the security appliance by including the strings **$(hostname)** and **$(domain)**. If you configure a banner in the system configuration, you can use that banner text within a context by using the **$(system)** string in the context configuration.

To add more than one line, precede each line by the **banner** command.

For example, to add a message-of-the-day banner, enter:

```
hostname(config)# banner motd Welcome to $(hostname).
hostname(config)# banner motd Contact me at admin@example.com for any
hostname(config)# banner motd issues.
```



CHAPTER 43

Managing Software and Configurations

This chapter contains information about managing the security appliance software and configurations, and includes the following sections:

- Viewing Files in Flash Memory, page 43-1
- Retrieving Files from Flash Memory, page 43-2
- Removing Files from Flash Memory, page 43-2
- Downloading Software or Configuration Files to Flash Memory, page 43-2
- Configuring the Application Image and ASDM Image to Boot, page 43-4
- Configuring the File to Boot as the Startup Configuration, page 43-5
- Performing Zero Downtime Upgrades for Failover Pairs, page 43-5
- Backing Up Configuration Files, page 43-8
- Configuring Auto Update Support, page 43-19

Viewing Files in Flash Memory

You can view files in Flash memory and see information about the files.

- To view the files in Flash memory, enter the following command:

  ```
  hostname# dir [flash: | disk0: | disk1:]
  ```

  The **flash:** keyword represents the internal Flash memory on the PIX 500 series security appliance. You can enter **flash:** or **disk0:** for the internal Flash memory on the ASA 5500 series adaptive security appliance. The **disk1:** keyword represents the external Flash memory on the ASA. The internal Flash memory is the default.

  For example:

  ```
  hostname# dir

  Directory of disk0:/
  500    -rw-  4958208     22:56:20 Nov 29 2004  cdisk.bin
  2513   -rw-  4634        19:32:48 Sep 17 2004  first-backup
  2788   -rw-  21601       20:51:46 Nov 23 2004  backup.cfg
  2927   -rw-  8670632     20:42:48 Dec 08 2004  asdmfile.bin
  ```

- To view extended information about a specific file, enter the following command:

  ```
  hostname# show file information [path:/]filename
  ```

The default path is the root directory of the internal Flash memory (flash:/ or disk0:/).

For example:

```
hostname# show file information cdisk.bin

disk0:/cdisk.bin:
  type is image (XXX) []
  file size is 4976640 bytes version 7.0(1)
```

The file size listed is for example only.

Retrieving Files from Flash Memory

You can retrieve files directly from the Flash disk by using an HTTPS connection with the following URL, in which you supply values for the ASA IP address and the filename:

`https://ASA_IP/disk0/filename`

This option is useful for customers who wish to do the following:

- Copyfrom the ASA binary image (as a backup).
- Copy from WebVPN capture files.
- Copy any other Flash files to a secure desktop.

Removing Files from Flash Memory

To delete a file in the disk partition from Flash memory, enter the following command:

```
hostname# delete flash:filename
```

By default, the file is deleted from the current working directory if you do not specify a path. You may use wildcards when deleting files. You are prompted with the filename to delete, and then you must confirm the deletion.

Downloading Software or Configuration Files to Flash Memory

You can download application images, ASDM images, configuration files, and other files to the internal Flash memory or, for the ASA 5500 series adaptive security appliance, to the external Flash memory from a TFTP, FTP, HTTP, or HTTPS server.

Note You cannot have two files with the same name but with different letter case in the same directory in Flash memory. For example,if you attempt to dowload the file Config.cfg to a location that contains the file config.cfg, you recieve the error `%Error opening disk0:/Config.cfg (File exists).`

This section includes the following topics:

- Downloading a File to a Specific Location, page 43-3
- Downloading a File to the Startup or Running Configuration, page 43-3

Downloading a File to a Specific Location

This section describes how to download the application image, ASDM software, a configuration file, or any other file that needs to be downloaded to Flash memory. To download a file to the running or startup configuration, see the "Downloading a File to the Startup or Running Configuration" section on page 43-3.

For information about installing the Cisco SSL VPN client, see the "Installing the AnyConnect Client" section on page 40-2. For information about installing Cisco Secure Desktop on the security appliance, see the *Cisco Secure Desktop Configuration Guide for Cisco ASA 5500 Series Administrators*.

To configure the security appliance to use a specific application image or ASDM image if you have more than one installed, or have installed them in external Flash memory see the "Configuring the Application Image and ASDM Image to Boot" section on page 43-4.

Note To successfully copy ASDM Version 6.0(1) to Flash memory, you must be running Version 8.0.

To configure the security appliance to use a specific configuration as the startup configuration, see the "Configuring the File to Boot as the Startup Configuration" section on page 43-5.

For multiple context mode, you must be in the system execution space.

To download a file to Flash memory, see the following commands for each download server type:

- To copy from a TFTP server, enter the following command:

  ```
  hostname# copy tftp://server[/path]/filename {flash:/ | disk0:/ |
  disk1:/}[path/]filename
  ```

  The **flash:/** keyword represents the internal Flash memory on the PIX 500 series security appliance. You can enter **flash:/** or **disk0:/** for the internal Flash memory on the ASA 5500 series adaptive security appliance. The **disk1:/** keyword represents the external Flash memory on the ASA.

- To copy from an FTP server, enter the following command:

  ```
  hostname# copy ftp://[user[:password]@]server[/path]/filename {flash:/ | disk0:/ |
  disk1:/}[path/]filename
  ```

- To copy from an HTTP or HTTPS server, enter the following command:

  ```
  hostname# copy http[s]://[user[:password]@]server[:port][/path]/filename {flash:/ |
  disk0:/ | disk1:/}[path/]filename
  ```

- To use secure copy, first enable SSH, then enter the following command:

  ```
  hostname# ssh scopy enable
  ```

Then from a Linux client enter the following command:

```
scp -v -pw password filename username@asa_address
```

The **-v** is for verbose, and if **-pw** is not specified you will be prompted for a password.

Downloading a File to the Startup or Running Configuration

You can download a text file to the running or startup configuration from a TFTP, FTP, or HTTP(S) server, or from the Flash memory.

To copy a file to the startup configuration or running configuration, enter one of the following commands for the appropriate download server.

Note When you copy a configuration to the running configuration, you merge the two configurations. A merge adds any new commands from the new configuration to the running configuration. If the configurations are the same, no changes occur. If commands conflict or if commands affect the running of the context, then the effect of the merge depends on the command. You might get errors, or you might have unexpected results.

- To copy from a TFTP server, enter the following command:

  ```
  hostname# copy tftp://server[/path]/filename {startup-config | running-config}
  ```

- To copy from an FTP server, enter the following command:

  ```
  hostname# copy ftp://[user[:password]@]server[/path]/filename {startup-config |
  running-config}
  ```

- To copy from an HTTP or HTTPS server, enter the following command:

  ```
  hostname# copy http[s]://[user[:password]@]server[:port][/path]/filename
  {startup-config | running-config}
  ```

- To copy from Flash memory, enter the following command:

  ```
  hostname# copy {flash:/ | disk0:/ | disk1:/}[path/]filename
  {startup-config | running-config}
  ```

For example, to copy the configuration from a TFTP server, enter the following command:

```
hostname# copy tftp://209.165.200.226/configs/startup.cfg startup-config
```

To copy the configuration from an FTP server, enter the following command:

```
hostname# copy ftp://admin:letmein@209.165.200.227/configs/startup.cfg startup-config
```

To copy the configuration from an HTTP server, enter the following command:

```
hostname# copy http://209.165.200.228/configs/startup.cfg startup-config
```

Configuring the Application Image and ASDM Image to Boot

By default, the security appliance boots the first application image it finds in internal Flash memory. It also boots the first ASDM image it finds in internal Flash memory, or of none exists there, then in external Flash memory. If you have more than one image, you should specify the image you want to boot. In the case of the ASDM image, if you do not specify the image to boot, even if you have only one image installed, then the security appliance inserts the **asdm image** command into the running configuration. To avoid problems with Auto Update (if configured), and to avoid the image search at each startup, you should specify the ASDM image you want to boot in the startup configuration.

- To configure the application image to boot, enter the following command:

  ```
  hostname(config)# boot system url
  ```

  where *url* is one of the following:

  – {**flash:/** | **disk0:/** | **disk1:/**}[*path*/]*filename*

The **flash:/** keyword represents the internal Flash memory on the PIX 500 series security appliance. You can enter **flash:/** or **disk0:/** for the internal Flash memory on the ASA 5500 series adaptive security appliance. The **disk1:/** keyword represents the external Flash memory on the ASA.

– **tftp://**[*user*[**:***password*]**@**]*server*[*:port*]**/**[*path/*]*filename*

This option is only supported for the ASA 5500 series adaptive security appliance.

You can enter up to four **boot system** command entries, to specify different images to boot from in order; the security appliance boots the first image it finds. Only one **boot system tftp** command can be configured, and it must be the first one configured.

Note If the adaptive security appliance is stuck in a cycle of contstant booting, you can reboot the security appliance into ROMMON mode. For more information about the ROMMON mode, see Using the ROM Monitor to Load a Software Image, page 45-10.

- To configure the ASDM image to boot, enter the following command:

```
hostname(config)# asdm image {flash:/ | disk0:/ | disk1:/}[path/]filename
```

Configuring the File to Boot as the Startup Configuration

By default, the security appliance boots from a startup configuration that is a hidden file. You can alternatively set any configuration to be the startup configuration by entering the following command:

```
hostname(config)# boot config {flash:/ | disk0:/ | disk1:/}[path/]filename
```

The **flash:/** keyword represents the internal Flash memory on the PIX 500 series security appliance. You can enter **flash:/** or **disk0:/** for the internal Flash memory on the ASA 5500 series adaptive security appliance. The **disk1:/** keyword represents the external Flash memory on the ASA.

Performing Zero Downtime Upgrades for Failover Pairs

The two units in a failover configuration should have the same major (first number) and minor (second number) software version. However, you do not need to maintain version parity on the units during the upgrade process; you can have different versions on the software running on each unit and still maintain failover support. To ensure long-term compatibility and stability, we recommend upgrading both units to the same version as soon as possible.

Table 43-1 shows the supported scenarios for performing zero-downtime upgrades on a failover pair.

Table 43-1 Zero-Downtime Upgrade Support

Type of Upgrade	Support
Maintenance Release	You can upgrade from any maintenance release to any other maintenance release within a minor release. For example, you can upgrade from 7.0(1) to 7.0(4) without first installing the maintenance releases in between.
Minor Release	You can upgrade from a minor release to the next minor release. You cannot skip a minor release. For example, you can upgrade from 7.0 to 7.1. Upgrading from 7.0 directly to 7.2 is not supported for zero-downtime upgrades; you must first upgrade to 7.1.
Major Release	You can upgrade from the last minor release of the previous version to the next major release. For example, you can upgrade from 7.9 to 8.0, assuming that 7.9 is the last minor version in the 7.x release.

For more details about upgrading the software on a failover pair, refer to the following topics:

- Upgrading an Active/Standby Failover Configuration, page 43-6
- Upgrading and Active/Active Failover Configuration, page 43-7

Upgrading an Active/Standby Failover Configuration

To upgrade two units in an Active/Standby failover configuration, perform the following steps:

Step 1 Download the new software to both units, and specify the new image to load with the **boot system** command (see the "Configuring the Application Image and ASDM Image to Boot" section on page 43-4).

Step 2 Reload the standby unit to boot the new image by entering the following command on the active unit:

```
active# failover reload-standby
```

Step 3 When the standby unit has finished reloading, and is in the Standby Ready state, force the active unit to fail over to the standby unit by entering the following command on the active unit.

Note Use the **show failover** command to verify that the standby unit is in the Standby Ready state.

```
active# no failover active
```

Step 4 Reload the former active unit (now the new standby unit) by entering the following command:

```
newstandby# reload
```

Step 5 When the new standby unit has finished reloading, and is in the Standby Ready state, return the original active unit to active status by entering the following command:

```
newstandby# failover active
```

Upgrading and Active/Active Failover Configuration

To upgrade two units in an Active/Active failover configuration, perform the following steps:

Step 1 Download the new software to both units, and specify the new image to load with the **boot system** command (see the "Configuring the Application Image and ASDM Image to Boot" section on page 43-4).

Step 2 Make both failover groups active on the primary unit by entering the following command in the system execution space of the primary unit:

```
primary# failover active
```

Step 3 Reload the secondary unit to boot the new image by entering the following command in the system execution space of the primary unit:

```
primary# failover reload-standby
```

Step 4 When the secondary unit has finished reloading, and both failover groups are in the Standby Ready state on that unit, make both failover groups active on the secondary unit using the following command in the system execution space of the primary unit:



Note Use the **show failover** command to verify that both failover groups are in the Standby Ready state on the secondary unit.

```
primary# no failover active
```

Step 5 Make sure both failover groups are in the Standby Ready state on the primary unit, and then reload the primary unit using the following command:

```
primary# reload
```

Step 6 If the failover groups are configured with the **preempt** command, they will automatically become active on their designated unit after the preempt delay has passed. If the failover groups are not configured with the **preempt** command, you can return them to active status on their designated units using the **failover active group** command.

Backing Up Configuration Files

To back up your configuration, use one of the following methods:

- Backing up the Single Mode Configuration or Multiple Mode System Configuration, page 43-8
- Backing Up a Context Configuration in Flash Memory, page 43-8
- Backing Up a Context Configuration within a Context, page 43-9
- Copying the Configuration from the Terminal Display, page 43-9

Backing up the Single Mode Configuration or Multiple Mode System Configuration

In single context mode or from the system configuration in multiple mode, you can copy the startup configuration or running configuration to an external server or to the local Flash memory:

- To copy to a TFTP server, enter the following command:

  ```
  hostname# copy {startup-config | running-config} tftp://server[/path]/filename
  ```

- To copy to a FTP server, enter the following command:

  ```
  hostname# copy {startup-config | running-config}
  ftp://[user[:password]@]server[/path]/filename
  ```

- To copy to local Flash memory, enter the following command:

  ```
  hostname# copy {startup-config | running-config} {flash:/ | disk0:/ |
  disk1:/}[path/]filename
  ```

  Be sure the destination directory exists. If it does not exist, first create the directory using the **mkdir** command.

  Backing up

Backing Up a Context Configuration in Flash Memory

In multiple context mode, copy context configurations that are on the local Flash memory by entering one of the following commands in the system execution space:

- To copy to a TFTP server, enter the following command:

  ```
  hostname# copy disk:[path/]filename tftp://server[/path]/filename
  ```

- To copy to a FTP server, enter the following command:

  ```
  hostname# copy disk:[path/]filename ftp://[user[:password]@]server[/path]/filename
  ```

- To copy to local Flash memory, enter the following command:

  ```
  hostname# copy {flash:/ | disk0:/ | disk1:/}[path/]filename {flash:/ | disk0:/ |
  disk1:/}[path/]newfilename
  ```

  Be sure the destination directory exists. If it does not exist, first create the directory using the **mkdir** command.

Backing Up a Context Configuration within a Context

In multiple context mode, from within a context, you can perform the following backups:

- To copy the running configuration to the startup configuration server (connected to the admin context), enter the following command:

  ```
  hostname/contexta# copy running-config startup-config
  ```

- To copy the running configuration to a TFTP server connected to the context network, enter the following command:

  ```
  hostname/contexta# copy running-config tftp:/server[/path]/filename
  ```

Copying the Configuration from the Terminal Display

To print the configuration to the terminal, enter the following command:

```
hostname# show running-config
```

Copy the output from this command, then paste the configuration in to a text file.

Backing Up Additional Files Using the Export and Import Commands

Additional files essential to your configuration might include the following:

- Files you import using the **import webvpn** command. Currently these files include customizations, URL lists, web contents, plug-ins, and language translations.
- DAP policies (dap.xml)
- CSD configurations (data.xml)
- Digital keys and certificates
- Local CA user database and certificate status files

The CLI lets you back up and restore individual elements of your configuration using the **export** and **import** commands. To back up these files, for example, those imported via the **import webvpn** command or certificates, follow these steps:

Step 1 Issue the appropriate **show** command(s). For example.

```
hostname # show import webvpn plug-in
ica
rdp
ssh,telnet
vnc
hostname#
```

Step 2 Issue the **export** command for the file you want to back up, in this example the rdp file.

```
hostname # export webvpn plug-in protocol rdp tftp://tftpserver/backupfilename
hostname #
```

Using a Script to Back Up and Restore Files

You can use a script to back up and restore the configuration files on your security appliance, including all of the extensions you import via the **import webvpn** CLI, the CSD configuration XML files, and the DAP configuration XML file. For security reasons, we do not recommend that you perform automated backups of digital keys and certificates or the Local CA key.

This section provides instructions for doing so, and includes a sample script that you can use as is or modify as your environment requires. The sample script is specific to a Linux system. To use it for a Microsoft Windows system, you need to modify it using the logic of the sample.

Note The existing CLI lets you back up and restore individual files using the **copy**, **export**, and **import** commands. It does not, however, have a facility that lets you back up all ASA configuration files in one operation. Running the script facilitates the use of multiple CLIs.

Prerequisites

To use a script to back up and restore an ASA configuration, first perform the following tasks:

- Install Perl with an Expect module.
- Install an SSH client that can reach the ASA.
- Install a TFTP server to send files from the ASA to the backup site.

Another option is to use a commercially available tool. You can put the logic of this script into such a tool.

Running the Script

To run a backup and restore script, follow these steps:

Step 1 Download or cut and paste the script file to any location on your system.

Step 2 At the command line, enter **Perl *scriptname***, where *scriptname* is the name of the script file.

Step 3 Press **Enter**.

Step 4 The system prompts you for values for each of the options. Alternatively, you can enter values for the options when you enter the **Perl *scriptname*** command before you press **Enter**. Either way, the script requires that you enter a value for each option.

Step 5 The script starts running, printing out the commands that it issues, which provides you with a record of the CLIs. You can use these CLIs for a later restore, particularly useful if you want to restore only one or two files.

Sample Script

```
#!/usr/bin/perl
#Function: Backup/restore configuration/extensions to/from a TFTP server.
#Description: The objective of this script is to show how to back up configurations/extensions
#       before the backup/restore command is developed.
#       It currently backs up the running configuration, all extensions imported via "import webvpn"
#       command, the CSD configuration XML file, and the DAP configuration XML file.
#Requirements: Perl with Expect, SSH to the ASA, and a TFTP server.
#Usage: backupasa -option option_value
#      -h: ASA hostname or IP address
#      -u: User name to log in via SSH
#      -w: Password to log in via SSH
#      -e: The Enable password on the security appliance
#      -p: Global configuration mode prompt
#      -s: Host name or IP address of the TFTP server to store the configurations
#      -r: Restore with an argument that specifies the the file name. This file is produced during backup.
#If you don't enter an option, the script will prompt for it prior to backup.
#
#Make sure that you can SSH to the ASA.

use Expect;
use Getopt::Std;

#global variables
%options=();
$restore = 0; #does backup by default
$restore_file = '';
$asa = '';
$storage = '';
$user = '';
$password = '';
$enable = '';
$prompt = '';
$date = `date +%F`;
chop($date);
my $exp = new Expect();
```

```
getopts("h:u:p:w:e:s:r:",\%options);
do process_options();

do login($exp);
do enable($exp);
if ($restore) {
  do restore($exp,$restore_file);
}
else {
  $restore_file = "$prompt-restore-$date.cli";
  open(OUT,">$restore_file") or die "Can't open $restore_file\n";
  do running_config($exp);
  do lang_trans($exp);
  do customization($exp);
  do plugin($exp);
  do url_list($exp);
  do webcontent($exp);
  do dap($exp);
  do csd($exp);
  close(OUT);
}
do finish($exp);

sub enable {
  $obj = shift;
  $obj->send("enable\n");
  unless ($obj->expect(15, 'Password:')) {
    print "timed out waiting for Password:\n";
  }
  $obj->send("$enable\n");
  unless ($obj->expect(15, "$prompt#")) {
    print "timed out waiting for $prompt#\n";
  }
}

sub lang_trans {
  $obj = shift;
  $obj->clear_accum();
```

```
  $obj->send("show import webvpn translation-table\n");
  $obj->expect(15, "$prompt#" );
  $output = $obj->before();
  @items = split(/\n+/, $output);

  for (@items) {
   s/^\s+//;
   s/\s+$//;
   next if /show import/ or /Translation Tables/;
   next unless (/^.+\s+.+$/);
   ($lang, $transtable) = split(/\s+/,$_);
   $cli = "export webvpn translation-table $transtable language $lang
$storage/$prompt-$date-$transtable-$lang.po";
   $ocli = $cli;
   $ocli =~ s/^export/import/;
   print "$cli\n";
   print OUT "$ocli\n";
   $obj->send("$cli\n");
   $obj->expect(15, "$prompt#" );
  }
}

sub running_config {
 $obj = shift;
 $obj->clear_accum();
 $cli ="copy /noconfirm running-config $storage/$prompt-$date.cfg";
 print "$cli\n";
 $obj->send("$cli\n");
 $obj->expect(15, "$prompt#" );
}


sub customization {
 $obj = shift;
 $obj->clear_accum();
 $obj->send("show import webvpn customization\n");
 $obj->expect(15, "$prompt#" );
 $output = $obj->before();
```

```
 @items = split(/\n+/, $output);

 for (@items) {
  chop;
  next if /^Template/ or /show import/ or /^\s*$/;
  $cli = "export webvpn customization $_ $storage/$prompt-$date-cust-$_.xml";
  $ocli = $cli;
  $ocli =~ s/^export/import/;
  print "$cli\n";
  print OUT "$ocli\n";
  $obj->send("$cli\n");
  $obj->expect(15, "$prompt#" );
 }
}

sub plugin {
  $obj = shift;
  $obj->clear_accum();
  $obj->send("show import webvpn plug-in\n");
  $obj->expect(15, "$prompt#" );
  $output = $obj->before();
  @items = split(/\n+/, $output);

  for (@items) {
   chop;
   next if /^Template/ or /show import/ or /^\s*$/;
   $cli = "export webvpn plug-in protocol $_ $storage/$prompt-$date-plugin-$_.jar";
   $ocli = $cli;
   $ocli =~ s/^export/import/;
   print "$cli\n";
   print OUT "$ocli\n";
   $obj->send("$cli\n");
   $obj->expect(15, "$prompt#" );
  }
}

sub url_list {
  $obj = shift;
```

```
  $obj->clear_accum();
  $obj->send("show import webvpn url-list\n");
  $obj->expect(15, "$prompt#" );
  $output = $obj->before();
  @items = split(/\n+/, $output);

  for (@items) {
   chop;
   next if /^Template/ or /show import/ or /^\s*$/ or /No bookmarks/;
   $cli="export webvpn url-list $_ $storage/$prompt-$date-urllist-$_.xml";
   $ocli = $cli;
   $ocli =~ s/^export/import/;
   print "$cli\n";
   print OUT "$ocli\n";
   $obj->send("$cli\n");
   $obj->expect(15, "$prompt#" );
  }
}

sub dap {
  $obj = shift;
  $obj->clear_accum();
  $obj->send("dir dap.xml\n");
  $obj->expect(15, "$prompt#" );

  $output = $obj->before();
  return 0 if($output =~ /Error/);

  $cli="copy /noconfirm dap.xml $storage/$prompt-$date-dap.xml";
  $ocli="copy /noconfirm $storage/$prompt-$date-dap.xml disk0:/dap.xml";
  print "$cli\n";
  print OUT "$ocli\n";
  $obj->send("$cli\n");
  $obj->expect(15, "$prompt#" );
}

sub csd {
  $obj = shift;
```

```
  $obj->clear_accum();
  $obj->send("dir sdesktop\n");
  $obj->expect(15, "$prompt#" );

  $output = $obj->before();
  return 0 if($output =~ /Error/);

  $cli="copy /noconfirm sdesktop/data.xml $storage/$prompt-$date-data.xml";
  $ocli="copy /noconfirm $storage/$prompt-$date-data.xml disk0:/sdesktop/data.xml";
  print "$cli\n";
  print OUT "$ocli\n";
  $obj->send("$cli\n");
  $obj->expect(15, "$prompt#" );
}

sub webcontent {
  $obj = shift;
  $obj->clear_accum();
  $obj->send("show import webvpn webcontent\n");
  $obj->expect(15, "$prompt#" );
  $output = $obj->before();
  @items = split(/\n+/, $output);

  for (@items) {
    s/^\s+//;
    s/\s+$//;
    next if /show import/ or /No custom/;
    next unless (/^.+\s+.+$/);
    ($url, $type) = split(/\s+/,$_);
    $turl = $url;
    $turl =~ s/\/\+//;
    $turl =~ s/\+\//-/;
    $cli = "export webvpn webcontent $url $storage/$prompt-$date-$turl";
    $ocli = $cli;
    $ocli =~ s/^export/import/;
    print "$cli\n";
    print OUT "$ocli\n";
    $obj->send("$cli\n");
```

```
    $obj->expect(15, "$prompt#" );
  }
}

sub login {
  $obj = shift;
  $obj->raw_pty(1);
  $obj->log_stdout(0); #turn off console logging.
  $obj->spawn("/usr/bin/ssh $user\@$asa") or die "can't spawn ssh\n";
  unless ($obj->expect(15, "password:" )) {
    die "timeout waiting for password:\n";
  }

  $obj->send("$password\n");

  unless ($obj->expect(15, "$prompt>" )) {
    die "timeout waiting for $prompt>\n";
  }
}

sub finish {
  $obj = shift;
  $obj->hard_close();
  print "\n\n";

}

sub restore {
  $obj = shift;
  my $file = shift;
  my $output;
  open(IN,"$file") or die "can't open $file\n";
  while (<IN>) {
    $obj->send("$_");
    $obj->expect(15, "$prompt#" );
    $output = $obj->before();
    print "$output\n";
  }
```

```
  close(IN);
}

sub process_options {
 if (defined($options{s})) {
    $tstr= $options{s};
    $storage = "tftp://$tstr";
 }
 else {
    print "Enter TFTP host name or IP address:";
    chop($tstr=<>);
    $storage = "tftp://$tstr";
 }
 if (defined($options{h})) {
    $asa = $options{h};
 }
 else {
    print "Enter ASA host name or IP address:";
    chop($asa=<>);
 }

 if (defined ($options{u})) {
    $user= $options{u};
 }
 else {
    print "Enter user name:";
    chop($user=<>);
 }

 if (defined ($options{w})) {
    $password= $options{w};
 }
 else {
    print "Enter password:";
    chop($password=<>);
 }
 if (defined ($options{p})) {
    $prompt= $options{p};
```

```
}
else {
   print "Enter ASA prompt:";
   chop($prompt=<>);
}
if (defined ($options{e})) {
   $enable = $options{e};
}
else {
   print "Enter enable password:";
   chop($enable=<>);
}

if (defined ($options{r})) {
  $restore = 1;
  $restore_file = $options{r};
}
}
```

Configuring Auto Update Support

Auto Update is a protocol specification that allows an Auto Update server to download configurations and software images to many security appliances, and can provide basic monitoring of the security appliances from a central location.

The security appliance can be configured as either a client or a server. As an Auto Update client, it periodically polls the Auto Update server for updates to software images and configuration files. As an Auto Update server, it issues updates for security appliances configured as Auto Update clients.

Note Auto Update is supported in single context mode only.

This section includes the following topics:

- Configuring Communication with an Auto Update Server, page 43-20
- Configuring Client Updates as an Auto Update Server, page 43-21
- Viewing Auto Update Status, page 43-22

Configuring Communication with an Auto Update Server

To configure the security appliance as an Auto Update client, perform the following steps:

Step 1 To specify the URL of the AUS, use the following command:

```
hostname(config)# auto-update server url [source interface] [verify-certificate]
```

Where *url* has the following syntax:

```
http[s]://[user:password@]server_ip[:port]/pathname
```

SSL is used when **https** is specified. The *user* and *password* arguments of the URL are used for Basic Authentication when logging in to the server. If you use the **write terminal**, **show configuration** or **show tech-support** commands to view the configuration, the user and password are replaced with '********'.

The default port is 80 for HTTP and 443 for HTTPS.

The **source** *interface* argument specifies which interface to use when sending requests to the AUS. If you specify the same interface specified by the **management-access** command, the Auto Update requests travel over the same IPSec VPN tunnel used for management access.

The **verify-certificate** keyword verifies the certificate returned by the AUS.

Step 2 (Optional) To identify the device ID to send when communicating with the AUS, enter the following command:

```
hostname(config)# auto-update device-id {hardware-serial | hostname | ipaddress [if-name]
| mac-address [if-name] | string text}
```

The identifier used is determined by using one of the following parameters:

- **hardware-serial**—Use the security appliance serial number.
- **hostname**—Use the security appliance hostname.
- **ipaddress**—Use the IP address of the specified interface. If the interface name is not specified, it uses the IP address of the interface used to communicate with the AUS.
- **mac-address**—Use the MAC address of the specified interface. If the interface name is not specified, it uses the MAC address of the interface used to communicate with the AUS.
- **string**—Use the specified text identifier, which cannot contain white space or the characters ‘, “, , >, & and ?.

Step 3 (Optional) To specify how often to poll the AUS for configuration or image updates, enter the following command:

```
hostname(config)# auto-update poll-period poll-period [retry-count [retry-period]]
```

The *poll-period* argument specifies how often (in minutes) to check for an update. The default is 720 minutes (12 hours).

The *retry-count* argument specifies how many times to try reconnecting to the server if the first attempt fails. The default is 0.

The *retry-period* argument specifies how long to wait (in minutes) between retries. The default is 5.

Step 4 (Optional) To schedule a specific time for the security appliance to poll the Auto Update server, use the following command:

```
hostname(config)# auto-update poll-at days-of-the-week time [randomize minutes] [retry_count
[retry_period]]
```

days-of-the-week is any single day or combination of days: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday and Sunday. Other possible values are daily (Monday through Sunday), weekdays (Monday through Friday) and weekend (Saturday and Sunday).

time specifies the time in the format HH:MM at which to start the poll. For example, 8:00 is 8:00 AM and 20:00 is 8:00 PM

randomize minutes specifies the period to randomize the poll time following the specified start time. The range is from 1 to 1439 minutes.

retry_count specifies how many times to try reconnecting to the Auto Update Server if the first attempt fails. The default is 0.

retry_period specifies how long to wait between connection attempts. The default is 5 minutes. The range is from 1 and 35791 minutes.

Step 5 (Optional) If the Auto Update Server has not been contacted for a certain period of time, the following command will cause it to cease passing traffic:

```
hostname(config)# auto-update timeout period
```

Where *period* specifies the timeout period in minutes between 1 and 35791. The default is to never time out (0). To restore the default, enter the **no** form of this command.

Use this command to ensure that the security appliance has the most recent image and configuration. This condition is reported with system log message 201008.

In the following example, a security appliance is configured to poll an AUS with IP address 209.165.200.224, at port number 1742, from the outside interface, with certificate verification.

It is also configured to use the hostname of the security appliance as the device ID. It is configured to poll every Friday and Saturday night at a random time between 10:00 p.m. and 11:00 p.m. On a failed polling attempt, it will try to reconnect to the AUS 10 times, and wait 3 minutes between attempts at reconnecting.

```
hostname(config)# auto-update server
https://jcrichton:farscape@209.165.200.224:1742/management source outside
verify-certificate
hostname(config)# auto-update device-id hostname
hostname(config)# auto-update poll-at Friday Saturday 22:00 randomize 60 2 10
```

Configuring Client Updates as an Auto Update Server

The **client-update** command lets you enable the update for security appliances configured as Auto Update clients. It lets you specify the type of software component (asdm or boot image), the type or family of security appliance, revision numbers to which the update applies, and a URL or IP address from which to get the update.

To configure the security appliance as an Auto Update server, perform the following steps:

Step 1 In global configuration mode, enable client update by entering the command:

```
hostname(config)# client-update enable
hostname(config)#
```

Step 2 Configure the parameters for the client update that you want to apply for the security appliances using the **client-update** command:

client-update {**component** {**asdm** | **image**} | **device-id** *dev_string* | **family** *family_name* | **type** *type*} **url** *url-string* **rev-nums** *rev-nums*}

component {**asdm** | **image**} specifies the software component, either ASDM or the boot image of the security appliance.

device-id *dev_string* specifies a unique string that the Auto Update client uses to identify itself. The maximum length is 63 characters.

family *family_name specifies the family name that the* Auto Update client uses to identify itself. It can be asa, pix, or a text string with a maximum length of 7 characters.

rev-nums *rev-nums* specifies the software or firmware images for this client. Enter up to 4, in any order, separated by commas.

type *type* specifies the type of clients to notify of a client update. Because this command is also used to update Windows clients, the list of clients includes several Windows operating systems. The security appliances in the list include the following:

- pix-515: Cisco PIX 515 Firewall
- pix-515e: Cisco PIX 515E Firewall
- pix-525: Cisco PIX 525 Firewall
- pix-535: Cisco PIX 535 Firewall
- asa5505: Cisco 5505 Adaptive Security Appliance
- asa5510: Cisco 5510 Adaptive Security Appliance
- asa5520: Cisco 5520 Adaptive Security Appliance
- asa5540: Cisco Adaptive Security Appliance

url *url-string* specifies the URL for the software/firmware image. This URL must point to a file appropriate for this client. For all Auto Update clients, you must use the protocol “http://” or “https://” as the prefix for the URL.

Configure the parameters for the client update that you want to apply to all security appliances of a particular type. That is, specify the type of security appliance and the URL or IP address from which to get the updated image. In addition, you must specify a revision number. If the revision number of the remote security appliance matches one of the specified revision numbers, there is no need to update—the client ignores the update.

The following example configures a client update for Cisco 5520 Adaptive Security Appliances:

```
hostname(config)# client-update type asa5520 component asdm url
http://192.168.1.114/aus/asdm601.bin rev-nums 8.0(1)
```

Viewing Auto Update Status

To view the Auto Update status, enter the following command:

```
hostname(config)# show auto-update
```

The following is sample output from the **show auto-update** command:

```
hostname(config)# show auto-update
Server: https://********@209.165.200.224:1742/management.cgi?1276
Certificate will be verified
Poll period: 720 minutes, retry count: 2, retry period: 5 minutes
Timeout: none
Device ID: host name [corporate]
Next poll in 4.93 minutes
Last poll: 11:36:46 PST Tue Nov 13 2004
Last PDM update: 23:36:46 PST Tue Nov 12 2004
```



CHAPTER 44

Monitoring the Security Appliance

This chapter describes how to monitor the security appliance, and includes the following sections:

- Using SNMP, page 44-1
- Configuring and Managing Logs, page 44-5

Using SNMP

This section describes how to use SNMP, and includes the following topics:

- SNMP Overview, page 44-1
- Enabling SNMP, page 44-4

SNMP Overview

The security appliance provides support for network monitoring using SNMP V1 and V2c. The security appliance supports traps and SNMP read access, but does not support SNMP write access.

You can configure the security appliance to send traps (event notifications) to an NMS, or you can use the NMS to browse the MIBs on the security appliance. MIBs are a collection of definitions, and the security appliance maintains a database of values for each definition. Browsing a MIB entails issuing an SNMP get request from the NMS. Use CiscoWorks for Windows or any other SNMP V1, MIB-II-compliant browser to receive SNMP traps and browse a MIB.

Table 44-1 lists supported MIBs and traps for the security appliance and, in multiple mode, for each context. You can download Cisco MIBs from the following website.

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

After you download the MIBs, compile them for your NMS.

Note In software versions 7.2(1), 8.0(2), and later, the SNMP information refreshes about every five seconds. As a result, we recommend that you wait for at least five seconds between consecutive polls.

Table 44-1 SNMP MIB and Trap Support

MIB or Trap Support	Description
SNMP core traps	The security appliance sends the following SNMP core traps: • authentication—An SNMP request fails because the NMS did not authenticate with the correct community string. • linkup—An interface has transitioned to the "up" state. • linkdown—An interface is down, for example, if you removed the **nameif** command. • coldstart—The security appliance is running after a reload.
SNMP link state traps	For the ASA 5505 only: • At bootup, the security appliance sends link state traps only on interfaces that were configured with a **nameif** command (that is, VLAN interfaces). Traps for physical interfaces (that is, Ethernet 0/0 and Ethernet 0/1) are also displayed. • When the Ethernet 0/1 interface is down, the security appliance sends traps about the two logical interfaces that are assigned to this physical interface. Traps for the logical and physical interfaces are displayed. • When the Ethernet 0/1 interface is up, the security appliance sends traps about the two logical interfaces that are assigned to this physical interface. Traps for the logical and physical interfaces are displayed.
MIB-II	The security appliance supports browsing of the following groups and tables: • system
IF-MIB	The security appliance supports browsing of the following tables: • ifTable • ifXTable For the ASA 5505 only: • All of the interfaces that are displayed with the internal interfaces are assigned an ifIndex, are displayed, and have their descriptions displayed. • Only the interfaces that have an assigned MTU have a value that is greater than zero. Use the **show interface details** command to validate the output. • The administrative status for all interfaces is displayed. • The operational status for all interfaces is displayed.
IP-MIB	For the ASA 5505 only: The output displays IP addresses that are assigned to the interfaces that were not configured using the **nameif** command.
RFC1213-MIB	The security appliance supports browsing of the following table: • ip.ipAddrTable

Table 44-1 SNMP MIB and Trap Support (continued)

MIB or Trap Support	Description
SNMPv2-MIB	The security appliance supports browsing the following: • snmp
ENTITY-MIB	The security appliance supports browsing of the following groups and tables: • entPhysicalTable • entLogicalTable The security appliance supports browsing of the following traps: • config-change • fru-insert • fru-remove
CISCO-IPSEC-FLOW-MONITOR-MIB	The security appliance supports browsing of the MIB. The security appliance supports browsing of the following traps: • start • stop
CISCO-REMOTE-ACCESS-MONITOR-MIB	The security appliance supports browsing of the MIB. The security appliance supports browsing of the following traps: • session-threshold-exceeded
CISCO-CRYPTO-ACCELERATOR-MIB	The security appliance supports browsing of the MIB.
ALTIGA-GLOBAL-REG	The security appliance supports browsing of the MIB.
Cisco Firewall MIB	The security appliance supports browsing of the following groups: • cfwSystem The information is cfwSystem.cfwStatus, which relates to failover status, pertains to the entire device and not just a single context. **Note** During failover, SNMP issues traps and syslog messages to notify you that the primary unit is down, and the secondary unit has become active. During the switchover event, the syslogs that were previously handling the linkup and link down events do not appear. For more information about failover, see the "Controlling and Monitoring Failover" section on page 15-51.
Cisco Memory Pool MIB	The security appliance supports browsing of the following table: • ciscoMemoryPoolTable—The memory usage described in this table applies only to the security appliance general-purpose processor.
Cisco Process MIB	The security appliance supports browsing of the following table: • cpmCPUTotalTable
Cisco Syslog MIB	The security appliance supports the following trap: • clogMessageGenerated You cannot browse this MIB.
CISCO-UNIFIED-FIREWALL-MIB	The security appliance supports browsing of the MIB.

Enabling SNMP

The SNMP agent that runs on the security appliance performs two functions:

- Replies to SNMP requests from NMSs.
- Sends traps (event notifications) to NMSs.

To enable the SNMP agent and identify an NMS that can connect to the security appliance, perform the following steps:

Step 1 Ensure that the SNMP server on the security appliance is enabled by entering the following command:

```
hostname(config)# snmp-server enable
```

The SNMP server is enabled by default.

Step 2 To identify the IP address of the NMS that can connect to the security appliance, enter the following command:

```
hostname(config)# snmp-server host interface_name ip_address [trap | poll] [community
text] [version 1 | 2c] [udp-port port]
```

Where *interface_name* is the name of the NMS and *ip_address* is the IP address of the NMS.

Specify **trap** or **poll** if you want to limit the NMS to receiving traps only or browsing (polling) only. By default, the NMS can use both functions.

SNMP traps are sent on UDP port 162 by default. You can change the port number using the **udp-port** keyword.

Step 3 To specify the community string, enter the following command:

```
hostname(config)# snmp-server community key
```

The SNMP community string is a shared secret between the security appliance and the NMS. The key is a case-sensitive value up to 32 characters in length. Spaces are not permitted.

Step 4 (Optional) To set the SNMP server location or contact information, enter the following command:

```
hostname(config)# snmp-server {contact | location} text
```

Where *text* defines the SNMP server location or lists contact information.

Step 5 To enable the security appliance to send traps to the NMS, enter the following command:

```
hostname(config)# snmp-server enable traps [all | syslog | snmp [trap] [...] |
entity [trap] [...] | ipsec [trap] [...] | remote-access [trap]]
```

Enter this command for each feature type to enable individual traps or sets of traps, or enter the **all** keyword to enable all traps.

The default configuration has all SNMP traps enabled (**snmp-server enable traps snmp authentication linkup linkdown coldstart**). You can disable these traps using the **no** form of this command with the **snmp** keyword. However, use the **clear configure snmp-server** command to restore the default enabling of SNMP traps.

If you enter this command and do not specify a trap type, then the default is the syslog trap. (The default SNMP traps continue to be enabled along with the syslog trap.)

SNMP traps include:

- **authentication**
- **linkup**

- **linkdown**
- **coldstart**

Entity traps include:

- **config-change**—The trigger for an SNMP configuration change trap is the creation or the deletion of a context.
- **fru-insert**
- **fru-remove**

IPSec traps include:

- **start**
- **stop**

Remote-access traps include:

- **session-threshold-exceeded**

Step 6 To enable system log messages to be sent as traps to the NMS, enter the following command:

```
hostname(config)# logging history level
```

Where *level* defines the logging severity level.

You must also enable syslog traps using the **snmp-server enable traps** command.

Step 7 To enable logging, so that system messages are generated and can then be sent to an NMS, enter the following command:

```
hostname(config)# logging enable
```

Note If SNMP traffic is not being allowed through the security appliance interfaces, you might also need to permit ICMP traffic from the remote SNMP server using the **icmp permit** command.

The following example sets the security appliance to receive requests from host 192.168.3.2 on the inside interface:

```
hostname(config)# snmp-server host 192.168.3.2
hostname(config)# snmp-server location building 42
hostname(config)# snmp-server contact Pat lee
hostname(config)# snmp-server community ohwhatakeyisthee
```

Configuring and Managing Logs

This section describes the logging functionality and configuration, as well as the system log message format, options, and variables. It includes the following topics:

- Logging Overview, page 44-6
- Logging in Multiple Context Mode, page 44-6
- Enabling and Disabling Logging, page 44-7
- Configuring Log Output Destinations, page 44-8
- Filtering System Log Messages, page 44-16

- Customizing the Log Configuration, page 44-19
- Understanding System Log Messages, page 44-24

Logging Overview

The security appliance system logs provide you with information for monitoring and troubleshooting the security appliance. With the logging feature, you can do the following:

- Specify which system log messages should be logged.
- Disable or change the severity level of a system log message.
- Specify one or more locations where system log messages should be sent, including an internal buffer, one or more syslog servers, ASDM, an SNMP management station, specified e-mail addresses, or to Telnet and SSH sessions.
- Size syslog servers according to the number of security appliances and other network devices that send syslog messages to the syslog server and to the number of syslog events that are generated per second by all devices.
- Configure and manage system log messages in groups, such as by severity level or class of message.
- Specify whether a rate-limit is applied to system log generation.
- Specify what happens to the contents of the internal buffer when the buffer becomes full: overwrite the buffer, send the buffer contents to an FTP server, or save the contents to internal Flash memory.

You can choose to send all system log messages, or subsets of system log messages, to any or all output locations. You can filter system log messages by locations, by the severity of the system log message, the class of the system log message, or by creating a custom system log message list.

Logging in Multiple Context Mode

Each security context includes its own logging configuration and generates its own messages. If you log in to the system or admin context, and then change to another context, messages you view in your session are only those that are related to the current context.

System log messages that are generated in the system execution space, including failover messages, are viewed in the admin context along with messages generated in the admin context. You cannot configure logging or view any logging information in the system execution space.

You can configure the security appliance to include the context name with each message, which helps you differentiate context messages that are sent to a single syslog server. This feature also helps you to determine which messages are from the admin context and which are from the system; messages that originate in the system execution space use a device ID of **system**, and messages that originate in the admin context use the name of the admin context as the device ID. For more information about enabling logging device IDs, see the "Including the Device ID in System Log Messages" section on page 44-20.

Analyzing Syslogs

The following are some examples of the type of information you can obtain from a review of various syslogs:

- Connections that are allowed by security appliance security policies. These messages help you spot "holes" that remain open in your security policies.
- Connections that are denied by security appliance security policies. These messages show what types of activity are being directed toward your secured inside network.

- Using the ACE deny rate logging feature shows attacks that are occurring against your security appliance.
- IDS activity messages can show attacks that have occurred.
- User authentication and command usage provide an audit trail of security policy changes.
- Bandwidth usage messages show each connection that was built and torn down, as well as the duration and traffic volume used.
- Protocol usage messages show the protocols and port numbers used for each connection.
- Address translation audit trail messages record NAT or PAT connections being built or torn down, which are useful if you receive a report of malicious activity coming from inside your network to the outside world.

For more information about analyzing syslogs, see the Taskflow for Configuring MARS to Monitor Security Appliances, page E-1.

Enabling and Disabling Logging

This section describes how to enable and disable logging on the security appliance and includes the following topics:

- Enabling Logging to All Configured Output Destinations, page 44-7
- Disabling Logging to All Configured Output Destinations, page 44-7
- Viewing the Log Configuration, page 44-7

Enabling Logging to All Configured Output Destinations

The following command enables logging; however, you must also specify at least one output destination so that you can view or save the logged messages. If you do not specify an output destination, the security appliance does not save system log messages generated when events occur.

For more information about configuring log output destinations, see the "Configuring Log Output Destinations" section on page 44-8.

To enable logging, enter the following command:

```
hostname(config)# logging enable
```

Disabling Logging to All Configured Output Destinations

To disable all logging to all configured log output destinations, enter the following command:

```
hostname(config)# no logging enable
```

Viewing the Log Configuration

To view the running log configuration, enter the following command:

```
hostname(config)# show logging
```

The example output of the **show logging** command is similar to the following:

```
Syslog logging: enabled
```

```
Facility: 16
Timestamp logging: disabled
Standby logging: disabled
Deny Conn when Queue Full: disabled
Console logging: disabled
Monitor logging: disabled
Buffer logging: disabled
Trap logging: level errors, facility 16, 3607 messages logged
    Logging to infrastructure 10.1.2.3
History logging: disabled
Device ID: 'inside' interface IP address "10.1.1.1"
Mail logging: disabled
ASDM logging: disabled
```

Configuring Log Output Destinations

This section describes how to specify where the security appliance should save or send the log messages that are generated and includes the following topics:

- Sending System Log Messages to a Syslog Server, page 44-8
- Sending System Log Messages to the Console Port, page 44-10
- Sending System Log Messages to an E-mail Address, page 44-10
- Sending System Log Messages to ASDM, page 44-11
- Sending System Log Messages to a Telnet or SSH Session, page 44-13
- Sending System Log Messages to the Log Buffer, page 44-14

Sending System Log Messages to a Syslog Server

This section describes how to configure the security appliance to send logs to a syslog server.

Configuring the security appliance to send logs to a syslog server enables you to archive logs, limited only by the available disk space on the server, and to manipulate log data after it is saved. For example, you could specify actions to be executed when certain types of system log messages are logged, extract data from the log and save the records to another file for reporting, or track statistics using a site-specific script.

To view logs generated by the security appliance, you must specify a log output destination. If you enable logging without specifying a log output destination, the security appliance generates messages, but does not save them to a location from which you can view them.

The syslog server must run a server program called “syslogd.” Windows (except for Windows 95 and Windows 98) provides a syslog server as part of its operating system. For Windows 95 and Windows 98, you must obtain a syslogd server from another vendor.

Note To start logging to a syslog server that you define in this procedure, be sure to enable logging for all output locations. See the “Enabling Logging to All Configured Output Destinations” section on page 44-7. To disable logging, see the “Disabling Logging to All Configured Output Destinations” section on page 44-7.

To configure the security appliance to send system log messages to a syslog server, perform the following steps:

Step 1 To designate a syslog server to receive the logs, enter the following command:

```
hostname(config)# logging host interface_name ip_address [tcp[/port] | udp[/port]]
[format emblem]
```

Where the **format emblem** keyword enables EMBLEM format logging for the syslog server (UDP only).

The *interface_name* argument specifies the interface through which you access the syslog server.

The *ip_address* argument specifies the IP address of the syslog server.

The **tcp[/***port***]** or **udp[/***port***]** argument specifies that the security appliance should use TCP or UDP to send system log messages to the syslog server. The default protocol is UDP. You can configure the security appliance to send data to a syslog server using either UDP or TCP, but not both. If you specify TCP, the security appliance discovers when the syslog server fails and discontinues sending logs. If you specify UDP, the security appliance continues to send logs regardless of whether the syslog server is operational. The *port* argument specifies the port that the syslog server listens to for system log messages. Valid port values are 1025 through 65535, for either protocol. The default UDP port is 514. The default TCP port is 1470.

For example:

```
hostname(config)# logging host dmz1 192.168.1.5
```

If you want to designate more than one syslog server as an output destination, enter a new command for each syslog server.

Step 2 To specify which system log messages should be sent to the syslog server, enter the following command:

```
hostname(config)# logging trap {severity_level | message_list}
```

Where the *severity_level* argument specifies the severity levels of messages to be sent to the syslog server. You can specify the severity level number (0 through 7) or name. For severity level names, see the "Severity Levels" section on page 44-25. For example, if you set the level to 3, then the security appliance sends system log messages for level 3, 2, 1, and 0.

The *message_list* argument specifies a customized message list that identifies the system log messages to send to the syslog server. For information about creating custom message lists, see the "Filtering System Log Messages with Custom Message Lists" section on page 44-18.

The following example specifies that the security appliance should send to the syslog server all system log messages with a severity level of level 3 (errors) and higher. The security appliance will send messages with the severity of 3, 2, and 1.

```
hostname(config)# logging trap errors
```

Step 3 (Optional) If needed, to continue TCP logging when the syslog server is down, enter the following command:

```
hostname(config)# logging host interface_name server_ip [tcp/port] [permit-hostdown]
```

Step 4 (Optional) If needed, set the logging facility to a value other than its default of 20 by entering the following command:

```
hostname(config)# logging facility number
```

Most UNIX systems expect the system log messages to arrive at facility 20.

Sending System Log Messages to the Console Port

This section describes how to configure the security appliance to send logs to the console port.

Note To start logging to the console port as defined in this procedure, be sure to enable logging for all output locations. See the "Enabling Logging to All Configured Output Destinations" section on page 44-7. To disable logging, see the "Disabling Logging to All Configured Output Destinations" section on page 44-7.

To specify which system log messages should be sent to the console port, enter the following command:

```
hostname(config)# logging console {severity_level | message_list}
```

Where the *severity_level* argument specifies the severity levels of messages to be sent to the console port. You can specify the severity level number (0 through 7) or name. For severity level names, see the "Severity Levels" section on page 44-25. For example, if you set the level to 3, then the security appliance sends system log messages for level 3, 2, 1, and 0.

The *message_list* argument specifies a customized message list that identifies the system log messages to send to the console port. For information about creating custom message lists, see the "Filtering System Log Messages with Custom Message Lists" section on page 44-18.

The following example specifies that the security appliance should send to the syslog server all system log messages with a severity level of level 3 (errors) and higher. The security appliance will send messages with the severity of 3, 2, and 1.

```
hostname(config)# logging console errors
```

Sending System Log Messages to an E-mail Address

You can configure the security appliance to send some or all system log messages to an e-mail address. When sent by e-mail, a system log message appears in the subject line of the e-mail message. For this reason, we recommend configuring this option to notify administrators of system log messages with high severity levels, such as critical, alert, and emergency.

Note To start logging to an e-mail address you define in this procedure, be sure to enable logging for all output locations. See the "Enabling Logging to All Configured Output Destinations" section on page 44-7. To disable logging, see the "Disabling Logging to All Configured Output Destinations" section on page 44-7.

To designate an e-mail address as an output destination, perform the following steps:

Step 1 To specify the system log messages to be sent to one or more e-mail addresses, enter the following command:

```
hostname(config)# logging mail {severity_level | message_list}
```

Where the *severity_level* argument specifies the severity levels of messages to be sent to the e-mail address. You can specify the severity level number (0 through 7) or name. For severity level names, see the "Severity Levels" section on page 44-25. For example, if you set the level to 3, then the security appliance sends system log messages for level 3, 2, 1, and 0.

The *message_list* argument specifies a customized message list that identifies the system log messages to send to the e-mail address. For information about creating custom message lists, see the "Filtering System Log Messages with Custom Message Lists" section on page 44-18.

The following example uses a *message_list* with the name "high-priority," previously set up with the **logging list** command:

```
hostname(config)# logging mail high-priority
```

Step 2 To specify the source e-mail address to be used when sending system log messages to an e-mail address, enter the following command:

```
hostname(config)# logging from-address email_address
```

For example:

```
hostname(config)# logging from-address xxx-001@example.com
```

Step 3 Specify the recipient e-mail address to be used when sending system log messages to an e-mail destination. You can configure up to five recipient addresses. You must enter each recipient separately.

To specify a recipient address, enter the following command:

```
hostname(config)# logging recipient-address e-mail_address [severity_level]
```

If a severity level is not specified, the default severity level is used (error condition, severity level 3).

For example:

```
hostname(config)# logging recipient-address admin@example.com
```

Step 4 To specify the SMTP server to be used when sending system log messages to an e-mail destination, enter the following command:

```
hostname(config)# smtp-server ip_address
```

For example:

```
hostname(config)# smtp-server 10.1.1.1
```

Sending System Log Messages to ASDM

You can configure the security appliance to send system log messages to ASDM. The security appliance sets aside a buffer area for system log messages waiting to be sent to ASDM and saves messages in the buffer as they occur. The ASDM log buffer is a different buffer than the internal log buffer. For information about the internal log buffer, see the "Sending System Log Messages to the Log Buffer" section on page 44-14.

When the ASDM log buffer is full, the security appliance deletes the oldest system log message to make room in the buffer for new system log messages. To control the number of system log messages retained in the ASDM log buffer, you can change the size of the buffer.

This section includes the following topics:

- Configuring Logging for ASDM, page 44-12
- Clearing the ASDM Log Buffer, page 44-13

Configuring Logging for ASDM

Note To start logging to ASDM as defined in this procedure, be sure to enable logging for all output locations. See the "Enabling Logging to All Configured Output Destinations" section on page 44-7. To disable logging, see the "Disabling Logging to All Configured Output Destinations" section on page 44-7.

To specify ASDM as an output destination, perform the following steps:

Step 1 To specify which system log messages should go to ASDM, enter the following command:

```
hostname(config)# logging asdm {severity_level | message_list}
```

Where the *severity_level* argument specifies the severity levels of messages to be sent to ASDM. You can specify the severity level number (0 through 7) or name. For severity level names, see the "Severity Levels" section on page 44-25. For example, if you set the level to 3, then the security appliance sends system log messages for level 3, 2, 1, and 0.

The *message_list* argument specifies a customized message list that identifies the system log messages to send to ASDM. For information about creating custom message lists, see the "Filtering System Log Messages with Custom Message Lists" section on page 44-18.

The following example shows how enable logging and send to the ASDM log buffer system log messages of severity levels 0, 1, and 2.

```
hostname(config)# logging asdm 2
```

Step 2 To specify the number of system log messages retained in the ASDM log buffer, enter the following command:

```
hostname(config)# logging asdm-buffer-size num_of_msgs
```

Where *num_of_msgs* specifies the number of system log messages that the security appliance retains in the ASDM log buffer.

The following example shows how to set the ASDM log buffer size to 200 system log messages.

```
hostname(config)# logging asdm-buffer-size 200
```

Configuring Secure Logging

Note You must use TCP only. Secure logging does not support UDP; an error occurs if you try to use this protocol.

To enable secure logging, enter the following command:

```
hostname(config)# logging host interface_name syslog_ip [tcp/port | udp/port] [format emblem]
[secure]
```

Where the *interface_name* argument specifies the interface on which the syslog server resides, the *syslog_ip* argument specifies the IP address of the syslog server, and the *port* argument specifies the port (TCP or UDP) that the syslog server listens to for messages.

The **tcp** keyword specifies that the security appliance should use TCP to send messages to the syslog server. The **udp** keyword specifies that the security appliance should use UDP to send messages to the syslog server. The **format emblem** keyword enables EMBLEM format logging for the syslog server. The **secure** keyword specifies that the connection to the remote logging host should use SSL/TLS for TCP only.

The following example shows how to set up secure logging:

```
hostname(config)# logging host inside 10.0.0.1 TCP/1500 secure
```

Clearing the ASDM Log Buffer

To erase the current contents of the ASDM log buffer, enter the following command:

```
hostname(config)# clear logging asdm
```

Sending System Log Messages to a Telnet or SSH Session

Viewing system log messages in a Telnet or SSH session requires two steps:

1. Specify which messages should be sent to Telnet or SSH sessions.
2. View logs in the current session.

This section includes the following topics:

- Configuring Logging for Telnet and SSH Sessions, page 44-13
- Viewing System Log Messages in the Current Session, page 44-13

Configuring Logging for Telnet and SSH Sessions

Note To start logging to a Telnet or SSH session as defined in this procedure, be sure to enable logging for all output locations. See the "Enabling Logging to All Configured Output Destinations" section on page 44-7. To disable logging, see the "Disabling Logging to All Configured Output Destinations" section on page 44-7.

To specify which messages should be sent to Telnet or SSH sessions, enter the following command:

```
hostname(config)# logging monitor {severity_level | message_list}
```

Where the *severity_level* argument specifies the severity levels of messages to be sent to the session. You can specify the severity level number (0 through 7) or name. For severity level names, see the "Severity Levels" section on page 44-25. For example, if you set the level to 3, then the security appliance sends system log messages for level 3, 2, 1, and 0.

The *message_list* argument specifies a customized message list that identifies the system log messages to send to the session. For information about creating custom message lists, see the "Filtering System Log Messages with Custom Message Lists" section on page 44-18.

Viewing System Log Messages in the Current Session

Step 1 After you log in to the security appliance, enable logging to the current session by entering the following command:

```
hostname# terminal monitor
```

This command enables logging only for the current session. If you log out, and then log in again, you need to reenter this command.

Step 2 To disable logging to the current session, enter the following command:

```
hostname(config)# terminal no monitor
```

Sending System Log Messages to the Log Buffer

If configured as an output destination, the log buffer serves as a temporary storage location for system log messages. New messages are appended to the end of the listing. When the buffer is full, that is, when the buffer wraps, old messages are overwritten as new messages are generated, unless you configure the security appliance to save the full buffer to another location.

This section includes the following topics:

- Enabling the Log Buffer as an Output Destination, page 44-14
- Viewing the Log Buffer, page 44-15
- Automatically Saving the Full Log Buffer to Flash Memory, page 44-15
- Automatically Saving the Full Log Buffer to an FTP Server, page 44-15
- Saving the Current Contents of the Log Buffer to Internal Flash Memory, page 44-16
- Clearing the Contents of the Log Buffer, page 44-16

Enabling the Log Buffer as an Output Destination

Note To start logging to the buffer as defined in this procedure, be sure to enable logging for all output locations. See the "Enabling Logging to All Configured Output Destinations" section on page 44-7. To disable logging, see the "Disabling Logging to All Configured Output Destinations" section on page 44-7.

To enable the log buffer as a log output destination, enter the following command:

```
hostname(config)# logging buffered {severity_level | message_list}
```

Where the *severity_level* argument specifies the severity levels of messages to be sent to the buffer. You can specify the severity level number (0 through 7) or name. For severity level names, see the "Severity Levels" section on page 44-25. For example, if you set the level to 3, then the security appliance sends system log messages for level 3, 2, 1, and 0.

The *message_list* argument specifies a customized message list that identifies the system log messages to send to the buffer. For information about creating custom message lists, see the "Filtering System Log Messages with Custom Message Lists" section on page 44-18.

For example, to specify that messages with severity levels 1 and 2 should be saved in the log buffer, enter one of the following commands:

```
hostname(config)# logging buffered critical
```

or

```
hostname(config)# logging buffered level 2
```

For the *message_list* option, specify the name of a message list containing criteria for selecting messages to be saved in the log buffer.

```
hostname(config)# logging buffered notif-list
```

Viewing the Log Buffer

To view the log buffer, enter the following command:

```
hostname(config)# show logging
```

Changing the Log Buffer Size

By default, the log buffer size is 4 KB. To change the size of the log buffer, enter the following command:

```
hostname(config)# logging buffer-size bytes
```

Where the *bytes* argument sets the amount of memory used for the log buffer, in bytes. For example, if you specify 8192, the security appliance uses 8 KB of memory for the log buffer.

The following example specifies that the security appliance uses 16 KB of memory for the log buffer:

```
hostname(config)# logging buffer-size 16384
```

Automatically Saving the Full Log Buffer to Flash Memory

Unless configured otherwise, the security appliance address messages to the log buffer on a continuing basis, overwriting old messages when the buffer is full. If you want to keep a history of logs, you can configure the security appliance to send the buffer contents to another output location each time the buffer fills. Buffer contents can be saved either to internal Flash memory or to an FTP server.

When saving the buffer content to another location, the security appliance creates log files with names that use a default time-stamp format, as follows:

```
LOG-YYYY-MM-DD-HHMMSS.TXT
```

where *YYYY* is the year, *MM* is the month, *DD* is the day of the month, and *HHMMSS* is the time in hours, minutes, and seconds.

While the security appliance writes the log buffer contents to internal Flash memory or an FTP server, the security appliance continues saving new messages to the log buffer.

To specify that messages in the log buffer should be saved to internal Flash memory each time the buffer wraps, enter the following command:

```
hostname(config)# logging flash-bufferwrap
```

Automatically Saving the Full Log Buffer to an FTP Server

See the "Saving the Current Contents of the Log Buffer to Internal Flash Memory" section for more information about saving the buffer.

To specify that messages in the log buffer should be saved to an FTP server each time the buffer wraps, perform the following steps.

Step 1 To enable the security appliance to send the log buffer contents to an FTP server every time the buffer wraps, enter the following command:

```
hostname(config)# logging ftp-bufferwrap
```

Step 2 To identify the FTP server, entering the following command:

```
hostname(config)# logging ftp-server server path username password
```

Where the *server* argument specifies the IP address of the external FTP server

The *path* argument specifies the directory path on the FTP server where the log buffer data is to be saved. This path is relative to the FTP root directory.

The *username* argument specifies a username that is valid for logging into the FTP server.

The *password* argument specifies the password for the username specified.

For example:

```
hostname(config)# logging ftp-server 10.1.1.1 /syslogs logsupervisor 1luvMy10gs
```

Saving the Current Contents of the Log Buffer to Internal Flash Memory

At any time, you can save the contents of the buffer to internal Flash memory. To save the current contents of the log buffer to internal Flash memory, enter the following command:

```
hostname(config)# logging savelog [savefile]
```

For example, the following example saves the contents of the log buffer to internal Flash memory using the file name latest-logfile.txt:

```
hostname(config)# logging savelog latest-logfile.txt
```

Clearing the Contents of the Log Buffer

To erase the contents of the log buffer, enter the following command:

```
hostname(config)# clear logging buffer
```

Filtering System Log Messages

This section describes how to specify which system log messages should go to output destinations, and includes the following topics:

- Message Filtering Overview, page 44-17
- Filtering System Log Messages by Class, page 44-17
- Filtering System Log Messages with Custom Message Lists, page 44-18

Message Filtering Overview

You can filter generated system log messages so that only certain system log messages are sent to a particular output destination. For example, you could configure the security appliance to send all system log messages to one output destination and to send a subset of those system log messages to a different output destination.

Specifically, you can configure the security appliance so that system log messages are directed to an output destination according to the following criteria:

- System log message ID number
- System log message severity level
- System log message class (equivalent to a functional area of the security appliance)

You customize these criteria by creating a message list that you can specify when you set the output destination in the "Configuring Log Output Destinations" section on page 44-8. Alternatively, you can configure the security appliance to send a particular message class to each type of output destination independently of the message list.

For example, you could configure the security appliance to send to the internal log buffer all system log messages with severity levels of 1, 2 and 3, send all system log messages in the "ha" class to a particular syslog server, or create a list of messages that you name "high-priority" that are sent to an e-mail address to notify system administrators of a possible problem.

Filtering System Log Messages by Class

The system log message class provides a method of categorizing system log messages by type, equivalent to a feature or function of the security appliance. For example, the "vpnc" class denotes the VPN client.

This section includes the following topics:

- Message Class Overview, page 44-17
- Sending All Messages in a Class to a Specified Output Destination, page 44-18

Message Class Overview

With logging classes, you can specify an output location for an entire category of system log messages with a single command.

You can use system log message classes in two ways:

- Issue the **logging class** command to specify an output location for an entire category of system log messages.
- Create a message list using the **logging list** command that specifies the message class. See the "Filtering System Log Messages with Custom Message Lists" section on page 44-18 for this method.

All system log messages in a particular class share the same initial three digits in their system log message ID numbers. For example, all system log message IDs that begin with the digits 611 are associated with the vpnc (VPN client) class. System log messages associated with the VPN client feature range from 611101 to 611323.

Sending All Messages in a Class to a Specified Output Destination

When you configure all messages in a class to go to a type of output destination, this configuration overrides the configuration in the specific output destination command. For example, if you specify that messages at level 7 should go to the log buffer, and you also specify that ha class messages at level 3 should go to the buffer, then the latter configuration takes precedence.

To configure the security appliance to send an entire system log message class to a configured output destination, enter the following command:

```
hostname(config)# logging class message_class {buffered | console | history | mail |
monitor | trap} [severity_level]
```

Where the *message_class* argument specifies a class of system log messages to be sent to the specified output destination.

The **buffered**, **history**, **mail**, **monitor**, and **trap** keywords specify the output destination to which system log messages in this class should be sent. The **history** keyword enables SNMP logging. The **monitor** keyword enables Telnet and SSH logging. The **trap** keyword enables syslog server logging. Select one destination per command line entry. If you want to specify that a class should go to more than one destination, enter a new command for each output destination.

The *severity_level* argument further restricts the system log messages to be sent to the output destination by specifying a severity level. For more information about message severity levels, see the "Severity Levels" section on page 44-25.

The following example specifies that all system log messages related to the class ha (high availability, also known as failover) with a severity level of 1 (alerts) should be sent to the internal logging buffer.

```
hostname(config)# logging class ha buffered alerts
```

For a list of the system log message classes and the ranges of system log message IDs associated with each class, see the *Cisco Security Appliance Logging Configuration and System Log Messages*.

Filtering System Log Messages with Custom Message Lists

Creating a custom message list is a flexible way to exercise fine control over which system log messages are sent to which output destination. In a custom system log message list, you specify groups of system log messages using any or all of the following criteria: severity level, message IDs, ranges of system log message IDs, or by message class.

For example, message lists can be used to do the following:

- Select system log messages with severity levels of 1 and 2 and send them to one or more e-mail addresses.
- Select all system log messages associated with a message class (such as "ha") and save them to the internal buffer.

A message list can include multiple criteria for selecting messages. However, you must add each message selection criteria with a new command entry. It is possible to create a message list containing overlapping message selection criteria. If two criteria in a message list select the same message, the message is logged only once.

To create a customized list that the security appliance can use to select messages to be saved in the log buffer, perform the following steps:

Step 1 Create a message list containing criteria for selecting messages by entering the following command:

```
hostname(config)# logging list name {level level [class message_class] |
message start_id[-end_id]}
```

Where the *name* argument specifies the name of the list. Do not use the names of severity levels as the name of a system log message list. Prohibited names include “emergencies,” “alert,” “critical,” “error,” “warning,” “notification,” “informational,” and “debugging.” Similarly, do not use the first three characters of these words at the beginning of a file name. For example, do not use a filename that starts with the characters “err.”

The **level** *level* argument specifies the severity level. You can specify the severity level number (0 through 7) or name. For severity level names, see the “Severity Levels” section on page 44-25. For example, if you set the level to 3, then the security appliance sends system log messages for level 3, 2, 1, and 0.

The **class** *message_class* argument specifies a particular message class.

The **message** *start_id*[**-***end_id*] argument specifies an individual system log message ID number or a range of numbers.

The following example creates a message list named notif-list that specifies messages with a severity level of 3 or higher should be saved in the log buffer:

```
hostname(config)# logging list notif-list level 3
```

Step 2 (Optional) If you want to add more criteria for message selection to the list, enter the same command as in the previous step, specifying the name of the existing message list and the additional criterion. Enter a new command for each criterion you want to add to the list.

The following example adds criteria to the message list—a range of message ID numbers and the message class ha (high availability or failover):

```
hostname(config)# logging list notif-list 104024-105999
hostname(config)# logging list notif-list level critical
hostname(config)# logging list notif-list level warning class ha
```

The preceding example states that system log messages that match the criteria specified will be sent to the output destination. The specified criteria for system log messages to be included in the list are the following:

- System log message IDs that fall in the range of 104024 to 105999
- All system log messages with critical level or higher (emergency, alert, or critical)
- All ha class system log messages with warning level or higher (emergency, alert, critical, error, or warning)

A system log message is logged if it satisfies any of these conditions. If a system log message satisfies more than one of the conditions, the message is logged only once.

Customizing the Log Configuration

This section describes other options for fine tuning the logging configuration and includes the following topics:

- Configuring the Logging Queue, page 44-20
- Including the Date and Time in System Log Messages, page 44-20
- Including the Device ID in System Log Messages, page 44-20

- Generating System Log Messages in EMBLEM Format, page 44-21
- Disabling a System Log Message, page 44-22
- Changing the Severity Level of a System Log Message, page 44-22
- Limiting the Rate of System Log Message Generation, page 44-23
- Changing the Amount of Internal Flash Memory Available for Logs, page 44-23

Configuring the Logging Queue

The security appliance has a fixed number of blocks in memory that can be allocated for buffering system log messages while they are waiting to be sent to the configured output destination. The number of blocks required depends on the length of the system log message queue and the number of syslog servers specified.

To specify the number of system log messages that the security appliance can hold in its queue before sending them to the configured output destination, enter the following command:

```
hostname(config)# logging queue message_count
```

Where the *message_count* variable specifies the number of system log messages that can remain in the system log message queue while awaiting processing. The default is 512 system log messages. A setting of 0 (zero) indicates unlimited system log messages, that is, the queue size is limited only by block memory availability.

To view the queue and queue statistics, enter the following command:

```
hostname(config)# show logging queue
```

Including the Date and Time in System Log Messages

To specify that system log messages should include the date and time that the system log messages was generated, enter the following command:

```
hostname(config)# logging timestamp
```

Including the Device ID in System Log Messages

To configure the security appliance to include a device ID in non-EMBLEM-format system log messages, enter the following command:

```
hostname(config)# logging device-id {context-name | hostname | ipaddress interface_name |
string text}
```

You can specify only one type of device ID for the system log messages.

The **context-name** keyword indicates that the name of the current context should be used as the device ID (applies to multiple context mode only). If you enable the logging device ID for the admin context in multiple context mode, messages that originate in the system execution space use a device ID of **system**, and messages that originate in the admin context use the name of the admin context as the device ID.

The **hostname** keyword specifies that the hostname of the security appliance should be used as the device ID.

The **ipaddress** *interface_name* argument specifies that the IP address of the interface specified as *interface_name* should be used as the device ID. If you use the **ipaddress** keyword, the device ID becomes the specified security appliance interface IP address, regardless of the interface from which the system log message is sent. This keyword provides a single, consistent device ID for all system log messages that are sent from the device.

The **string** *text* argument specifies that the text string should be used as the device ID. The string can contain as many as 16 characters. You cannot use blank spaces or any of the following characters:

- & (ampersand)
- ' (single quote)
- " (double quote)
- < (less than)
- > (greater than)
- ? (question mark)

Note If enabled, the device ID does not appear in EMBLEM-formatted system log messages or SNMP traps.

The following example enables the logging device ID for the security appliance:

```
hostname(config)# logging device-id hostname
```

The following example enables the logging device ID for a security context on the security appliance:

```
hostname(config)# logging device-id context-name
```

Generating System Log Messages in EMBLEM Format

To use the EMBLEM format for system log messages sent to destinations other than a syslog server, enter the following command:

```
hostname(config)# logging emblem
```

To use the EMBLEM format for system log messages sent to a syslog server over UDP, specify the **format emblem** option when you configure the syslog server as an output destination. Enter the following command:

```
hostname(config)# logging host interface_name ip_address {tcp[/port] | udp[/port]}
[format emblem]
```

Where the *interface_name* and *ip_address* specifies the syslog server to receive the system log messages, **tcp**[/*port*] and **udp**[/*port*] indicate the protocol and port that should be used, and **format emblem** enables EMBLEM formatting for messages sent to the syslog server.

The security appliance can send system log messages using either the UDP or TCP protocol; however, you can enable the EMBLEM format only for messages sent over UDP. The default protocol and port are UDP and 514.

For example:

```
hostname(config)# logging host interface_1 122.243.006.123 udp format emblem
```

For more information about syslog servers, see the "Sending System Log Messages to a Syslog Server" section on page 44-8.

Disabling a System Log Message

To prevent the security appliance from generating a particular system log message, enter the following command:

```
hostname(config)# no logging message message_number
```

For example:

```
hostname(config)# no logging message 113019
```

To reenable a disabled system log message, enter the following command:

```
hostname(config)# logging message message_number
```

For example:

```
hostname(config)# logging message 113019
```

To see a list of disabled system log messages, enter the following command:

```
hostname(config)# show logging message
```

To reenable logging of all disabled system log messages, enter the following command:

```
hostname(config)# clear config logging disabled
```

Changing the Severity Level of a System Log Message

To specify the logging level of a system log message, enter the following command:

```
hostname(config)# logging message message_ID level severity_level
```

The following example modifies the severity level of system log message ID 113019 from 4 (warnings) to 5 (notifications):

```
hostname(config)# logging message 113019 level 5
```

To reset the logging level of a system log message to its default level, enter the following command.

```
hostname(config)# no logging message message_ID level current_severity_level
```

The following example modifies the severity level of system log message ID 113019 to its default value of 4 (warnings):

```
hostname(config)# no logging message 113019 level 5
```

To see the severity level of a specific message, enter the following command:

```
hostname(config)# show logging message message_ID
```

To see a list of system log messages with modified severity levels, enter the following command:

```
hostname(config)# show logging message
```

To reset the severity level of all modified system log messages back to their defaults, enter the following command:

```
hostname(config)# clear configure logging level
```

The following example shows the use of the **logging message** command to control both whether a system log message is enabled and the severity level of the system log message:

```
hostname(config)# show logging message 403503
```

```
syslog 403503: default-level errors (enabled)

hostname(config)# logging message 403503 level 1
hostname(config)# show logging message 403503
syslog 403503: default-level errors, current-level alerts (enabled)

hostname(config)# no logging message 403503
hostname(config)# show logging message 403503
syslog 403503: default-level errors, current-level alerts (disabled)

hostname(config)# logging message 403503
hostname(config)# show logging message 403503
syslog 403503: default-level errors, current-level alerts (enabled)

hostname(config)# no logging message 403503 level 3
hostname(config)# show logging message 403503
syslog 403503: default-level errors (enabled)
```

Limiting the Rate of System Log Message Generation

The logging rate-limit determines the rate at which system log messages are generated.You can specify the rate at which system log messages are generated by applying a specified severity level (1 through 7) to a set of messages or to an individual message within a specified time period.

To limit the logging rate-limit, enter the following command:

```
hostname(config)# logging rate-limit
```

To show the disallowed system log messages, enter the following command:

```
hostname(config)# show logging rate-limit
```

To show the current logging rate-limit setting, enter the following command:

```
hostname(config)# show running-config logging rate-limit
```

To reset the logging rate-limit to the default value, enter the following command:

```
hostname(config)# clear running-config logging rate-limit
```

To reset the logging rate-limit, enter the following command:

```
hostname(config)# clear configure logging rate-limit
```

Changing the Amount of Internal Flash Memory Available for Logs

You can cause the security appliance to save the contents of the log buffer to internal Flash memory in two ways:

- Configure logging so that the contents of the log buffer are saved to internal Flash memory each time the buffer wraps
- Enter a command instructing the security appliance to save the current contents of the log buffer to internal Flash memory immediately

By default, the security appliance can use up to 1 MB of internal Flash memory for log data. The default minimum amount of internal Flash memory that must be free for the security appliance to save log data is 3 MB.

If a log file being saved to internal Flash memory would cause the amount of free internal Flash memory to fall below the configured minimum limit, the security appliance deletes the oldest log files to ensure that the minimum amount of memory remains free after saving the new log file. If there are no files to delete or if, after all old files are deleted, free memory would still be below the limit, the security appliance fails to save the new log file.

To modify the settings for the amount of internal Flash memory available for logs, perform the following steps:

Step 1 To specify the maximum amount of internal Flash memory available for saving log files, enter the following command:

```
hostname(config)# logging flash-maximum-allocation kbytes
```

Where *kbytes* specifies the maximum amount of internal Flash memory, in kilobytes, that can be used for saving log files.

The following example sets the maximum amount of internal Flash memory that can be used for log files to approximately 1.2 MB:

```
hostname(config)# logging flash-maximum-allocation 1200
```

Step 2 To specify the minimum amount of internal Flash memory that must be free for the security appliance to save a log file, enter the following command:

```
hostname(config)# logging flash-minimum-free kbytes
```

Where *kbytes* specifies the minimum amount of internal Flash memory, in kilobytes, that must be available before the security appliance saves a new log file.

The following example specifies that the minimum amount of free internal Flash memory must be 4000 KB before the security appliance can save a new log file:

```
hostname(config)# logging flash-minimum-free 4000
```

Understanding System Log Messages

This section describes the contents of system log messages generated by the security appliance. It includes the following topics:

- System Log Message Format, page 44-24
- Severity Levels, page 44-25

System Log Message Format

System log messages begin with a percent sign (%) and are structured as follows:

```
%PIX|ASA Level Message_number: Message_text
```

Field descriptions are as follows:

PIX|ASA	Identifies the system log message facility code for messages generated by the security appliance. This value is always PIX|ASA.
Level	1-7. The level reflects the severity of the condition described by the system log message. The lower the number, the more severe the condition. See Table 44-2 for more information.
Message_number	A unique six-digit number that identifies the system log message.
Message_text	A text string describing the condition. This portion of the system log message sometimes includes IP addresses, port numbers, or usernames.

Severity Levels

Table 44-2 lists the system log message severity levels.

Table 44-2 System Log Message Severity Levels

Level Number	Level Keyword	Description
0	**emergencies**	System unusable.
1	**alert**	Immediate action needed.
2	**critical**	Critical condition.
3	**error**	Error condition.
4	**warning**	Warning condition.
5	**notification**	Normal but significant condition.
6	**informational**	Informational message only.
7	**debugging**	Appears during debugging only.

Note The security appliance does not generate system log messages with a severity level of 0 (emergencies). This level is provided in the **logging** command for compatibility with the UNIX system log feature, but is not used by the security appliance.



CHAPTER 45

Troubleshooting the Security Appliance

This chapter describes how to troubleshoot the security appliance, and includes the following sections:

- Testing Your Configuration, page 45-1
- Reloading the Security Appliance, page 45-6
- Performing Password Recovery, page 45-6
- Using the ROM Monitor to Load a Software Image, page 45-10
- Erasing the Flash File System, page 45-12
- Other Troubleshooting Tools, page 45-12
- Common Problems, page 45-13

Testing Your Configuration

This section describes how to test connectivity for the single mode security appliance or for each security context, how to ping the security appliance interfaces, and how to allow hosts on one interface to ping through to hosts on another interface.

We recommend that you only enable pinging and debug messages during troubleshooting. When you are done testing the security appliance, follow the steps in "Disabling the Test Configuration" section on page 45-5.

This section includes the following topics:

- Enabling ICMP Debug Messages and System Log Messages, page 45-1
- Pinging Security Appliance Interfaces, page 45-2
- Pinging Through the Security Appliance, page 45-4
- Disabling the Test Configuration, page 45-5

Enabling ICMP Debug Messages and System Log Messages

Debug messages and system log messages can help you troubleshoot why your pings are not successful. The security appliance only shows ICMP debug messages for pings to the security appliance interfaces, and not for pings through the security appliance to other hosts. To enable debugging and system log messages, perform the following steps:

Step 1 To show ICMP packet information for pings to the security appliance interfaces, enter the following command:

```
hostname(config)# debug icmp trace
```

Step 2 To set system log messages to be sent to Telnet or SSH sessions, enter the following command:

```
hostname(config)# logging monitor debug
```

You can alternately use the **logging buffer debug** command to send log messages to a buffer, and then view them later using the **show logging** command.

Step 3 To send the system log messages to a Telnet or SSH session, enter the following command:

```
hostname(config)# terminal monitor
```

Step 4 To enable system log messages, enter the following command:

```
hostname(config)# logging on
```

The following example shows a successful ping from an external host (209.165.201.2) to the security appliance outside interface (209.165.201.1):

```
hostname(config)# debug icmp trace
Inbound ICMP echo reply (len 32 id 1 seq 256) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 512) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 512) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 768) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 768) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 1024) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 1024) 209.165.201.1 > 209.165.201.2
```

This example shows the ICMP packet length (32 bytes), the ICMP packet identifier (1), and the ICMP sequence number (the ICMP sequence number starts at 0 and is incremented each time that a request is sent).

Pinging Security Appliance Interfaces

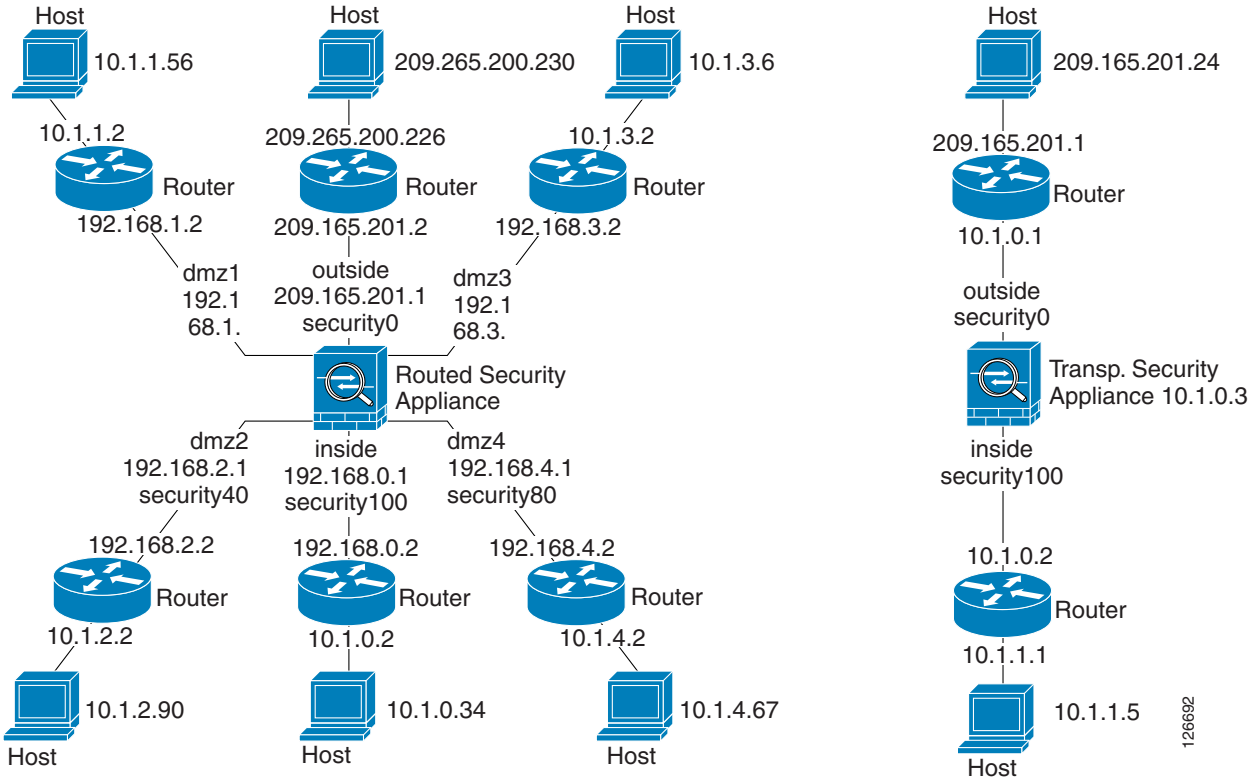
To test whether the security appliance interfaces are up and running and that the security appliance and connected routers are operating correctly, you can ping the security appliance interfaces. To ping the security appliance interfaces, perform the following steps:

Step 1 Draw a diagram of your single-mode security appliance or security context that shows the interface names, security levels, and IP addresses.

Note Although this procedure uses IP addresses, the **ping** command also supports DNS names and names that are assigned to a local IP address with the **name** command.

The diagram should also include any directly connected routers, and a host on the other side of the router from which you will ping the security appliance. You will use this information in this procedure and in the procedure in "Pinging Through the Security Appliance" section on page 45-4. For example:

Figure 45-1 Network Diagram with Interfaces, Routers, and Hosts



Step 2 Ping each security appliance interface from the directly connected routers. For transparent mode, ping the management IP address. This test ensures that the security appliance interfaces are active and that the interface configuration is correct.

A ping might fail if the security appliance interface is not active, the interface configuration is incorrect, or if a switch between the security appliance and a router is down (see Figure 45-2). In this case, no debug messages or system log messages appear, because the packet never reaches the security appliance.

Figure 45-2 Ping Failure at Security Appliance Interface

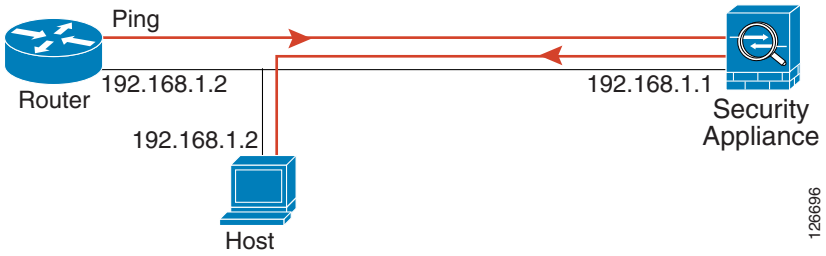


If the ping reaches the security appliance, and the security appliance responds, debug messages similar to the following appear:

```
ICMP echo reply (len 32 id 1 seq 256) 209.165.201.1 > 209.165.201.2
ICMP echo request (len 32 id 1 seq 512) 209.165.201.2 > 209.165.201.1
```

If the ping reply does not return to the router, then a switch loop or redundant IP addresses may exist (see Figure 45-3).

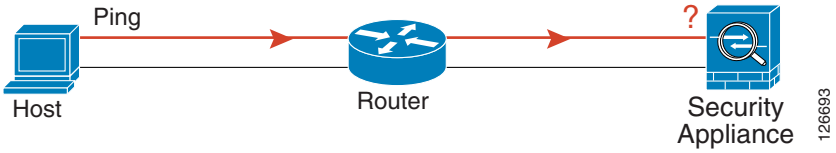
Figure 45-3 Ping Failure Because of IP Addressing Problems



Step 3 Ping each security appliance interface from a remote host. For transparent mode, ping the management IP address. This test checks whether the directly connected router can route the packet between the host and the security appliance, and whether the security appliance can correctly route the packet back to the host.

A ping might fail if the security appliance does not have a return route to the host through the intermediate router (see Figure 45-4). In this case, the debug messages show that the ping was successful, but system log message 110001 appears, indicating a routing failure.

Figure 45-4 Ping Failure Because the Security Appliance has No Return Route



Pinging Through the Security Appliance

After you successfully ping the security appliance interfaces, make sure traffic can pass successfully through the security appliance. For routed mode, this test shows that NAT is operating correctly, if configured. For transparent mode, which does not use NAT, this test confirms that the security appliance is operating correctly. If the ping fails in transparent mode, contact Cisco TAC.

To ping between hosts on different interfaces, perform the following steps:

Step 1 To add an access list allowing ICMP from any source host, enter the following command:

```
hostname(config)# access-list ICMPACL extended permit icmp any any
```

By default, when hosts access a lower security interface, all traffic is allowed through. However, to access a higher security interface, you need the preceding access list.

Step 2 To assign the access list to each source interface, enter the following command:

```
hostname(config)# access-group ICMPACL in interface interface_name
```

Repeat this command for each source interface.

Step 3 To enable the ICMP inspection engine and ensure that ICMP responses may return to the source host, enter the following commands:

```
hostname(config)# class-map ICMP-CLASS
hostname(config-cmap)# match access-list ICMPACL
```

```
hostname(config-cmap)# policy-map ICMP-POLICY
hostname(config-pmap)# class ICMP-CLASS
hostname(config-pmap-c)# inspect icmp
hostname(config-pmap-c)# service-policy ICMP-POLICY global
```

Alternatively, you can also apply the ICMP access list to the destination interface to allow ICMP traffic back through the security appliance.

Step 4 Ping from the host or router through the source interface to another host or router on another interface.

Repeat this step for as many interface pairs as you want to check.

If the ping succeeds, a system log message appears to confirm the address translation for routed mode (305009 or 305011) and that an ICMP connection was established (302020). You can also enter either the **show xlate** or **show conns** command to view this information.

If the ping fails for transparent mode, contact Cisco TAC.

For routed mode, the ping might fail because NAT is not configured correctly (see Figure 45-5). This failure is more likely to occur if you enable NAT control. In this case, a system log message appears, showing that the NAT failed (305005 or 305006). If the ping is from an outside host to an inside host, and you do not have a static translation (required with NAT control), the following system log message appears: "106010: deny inbound icmp."

Note The security appliance only shows ICMP debug messages for pings to the security appliance interfaces, and not for pings through the security appliance to other hosts.

Figure 45-5 Ping Failure Because the Security Appliance is not Translating Addresses



Disabling the Test Configuration

After you complete your testing, disable the test configuration that allows ICMP to and through the security appliance and that prints debug messages. If you leave this configuration in place, it can pose a serious security risk. Debug messages also slow the security appliance performance.

To disable the test configuration, perform the following steps:

Step 1 To disable ICMP debug messages, enter the following command:

```
hostname(config)# no debug icmp trace
```

Step 2 To disable logging, if desired, enter the following command:

```
hostname(config)# no logging on
```

Step 3 To remove the ICMPACL access list, and delete the related **access-group** commands, enter the following command:

```
hostname(config)# no access-list ICMPACL
```

Step 4 (Optional) To disable the ICMP inspection engine, enter the following command:

```
hostname(config)# no service-policy ICMP-POLICY
```

Traceroute

You can trace the route of a packet using the traceroute feature, which is accessed with the **traceroute** command. A traceroute works by sending UDP packets to a destination on an invalid port. Because the port is not valid, the routers along the way to the destination respond with an ICMP Time Exceeded Message, and report that error to the security appliance.

Packet Tracer

In addition, you can trace the lifespan of a packet through the security appliance to see whether the packet is operating correctly with the packet tracer tool. This tool lets you do the following:

- Debug all packet drops in a production network.
- Verify the configuration is working as intended.
- Show all rules applicable to a packet, along with the CLI commands that caused the rule addition.
- Show a time line of packet changes in a data path.
- Inject tracer packets into the data path.

The **packet-tracer** command provides detailed information about the packets and how they are processed by the security appliance. If a command from the configuration did not cause the packet to drop, the **packet-tracer** command will provide information about the cause in an easily readable manner. For example, when a packet is dropped because of an invalid header validation, the following message appears: “packet dropped due to bad ip header (reason).”

Reloading the Security Appliance

In multiple mode, you can only reload from the system execution space. To reload the security appliance, enter the following command:

```
hostname# reload
```

Performing Password Recovery

This section describes how to recover passwords if you have forgotten them or you are locked out because of AAA settings, and how to disable password recovery for extra security. This section includes the following topics:

- Recovering Passwords for the ASA 5500 Series Adaptive Security Appliance, page 45-7
- Recovering Passwords for the PIX 500 Series Security Appliance, page 45-8
- Disabling Password Recovery, page 45-9

- Resetting the Password on the SSM Hardware Module, page 45-10

Recovering Passwords for the ASA 5500 Series Adaptive Security Appliance

To recover passwords for the ASA 5500 Series adaptive security appliance, perform the following steps:

Step 1 Connect to the adaptive security appliance console port according to the instructions in "Accessing the Command-Line Interface" section on page 2-4.

Step 2 Power off the adaptive security appliance, and then power it on.

Step 3 After startup, press the **Escape** key when you are prompted to enter ROMMON mode.

Step 4 To update the configuration register value, enter the following command:

```
rommon #1> confreg 0x41
Update Config Register (0x41) in NVRAM...
```

Step 5 To set the adaptive security appliance to ignore the startup configuration, enter the following command:

```
rommon #1> confreg
```

The adaptive security appliance displays the current configuration register value, and asks whether you want to change it:

```
Current Configuration Register: 0x00000041
Configuration Summary:
  boot default image from Flash
  ignore system configuration

Do you wish to change this configuration? y/n [n]: y
```

Step 6 Record the current configuration register value, so you can restore it later.

Step 7 At the prompt, enter **Y** to change the value.

The adaptive security appliance prompts you for new values.

Step 8 Accept the default values for all settings. At the prompt, enter **Y**.

Step 9 Reload the adaptive security appliance by entering the following command:

```
rommon #2> boot
Launching BootLoader...
Boot configuration file contains 1 entry.

Loading disk0:/asa800-226-k8.bin... Booting...Loading...
```

The adaptive security appliance loads the default configuration instead of the startup configuration.

Step 10 Access the privileged EXEC mode by entering the following command:

```
hostname> enable
```

Step 11 When prompted for the password, press **Enter**.

The password is blank.

Step 12 Access the global configuration mode by entering the following command:

```
hostname# configure terminal
```

Step 13 Copy the running configuration to the startup configuration by entering the following command:

```
hostname(config)# copy running-config startup-config
```

Step 14 Change the passwords, as required, in the default configuration by entering the following commands:

```
hostname(config)# password password
hostname(config)# enable password password
hostname(config)# username name password password
```

Step 15 Load the default configuration by entering the following command:

```
hostname(config)# no config-register
```

The default configuration register value is 0x1. For more information about the configuration register, see the *Cisco Security Appliance Command Reference*.

Step 16 Save the new passwords to the startup configuration by entering the following command:

```
hostname(config)# copy running-config startup-config
```

Recovering Passwords for the PIX 500 Series Security Appliance

Recovering passwords on the PIX 500 Series security appliance erases the login password, enable password, and **aaa authentication console** commands. To recover passwords for the PIX 500 Series security appliance, perform the following steps:

Step 1 Download the PIX password tool from Cisco.com to a TFTP server accessible from the security appliance. For instructions, go to the following URL:

http://www.cisco.com/en/US/products/hw/vpndevc/ps2030/products_password_recovery09186a008009478b.shtml

Step 2 Connect to the security appliance console port according to the instructions in "Accessing the Command-Line Interface" section on page 2-4.

Step 3 Power off the security appliance, and then power it on.

Step 4 Immediately after the startup messages appear, press the **Escape** key to enter monitor mode.

Step 5 In monitor mode, configure the interface network settings to access the TFTP server by entering the following commands:

```
monitor> interface interface_id
monitor> address interface_ip
monitor> server tftp_ip
monitor> file pw_tool_name
monitor> gateway gateway_ip
```

Step 6 Download the PIX password tool from the TFTP server by entering the following command:

```
monitor> tftp
```

If you have trouble reaching the server, enter the **ping** *address* command to test the connection.

Step 7 At the "Do you wish to erase the passwords?" prompt, enter **Y**.

You can log in with the default login password of "cisco" and the blank enable password.

The following example shows password recovery on a PIX 500 Series security appliance with the TFTP server on the outside interface:

```
monitor> interface 0
0: i8255X @ PCI(bus:0 dev:13 irq:10)
1: i8255X @ PCI(bus:0 dev:14 irq:7 )

Using 0: i82559 @ PCI(bus:0 dev:13 irq:10), MAC: 0050.54ff.82b9
monitor> address 10.21.1.99
address 10.21.1.99
monitor> server 172.18.125.3
server 172.18.125.3
monitor> file np70.bin
file np52.bin
monitor> gateway 10.21.1.1
gateway 10.21.1.1
monitor> ping 172.18.125.3
Sending 5, 100-byte 0xf8d3 ICMP Echoes to 172.18.125.3, timeout is 4 seconds:
!!!!!
Success rate is 100 percent (5/5)
monitor> tftp
tftp np52.bin@172.18.125.3 via 10.21.1.1
Received 73728 bytes

Cisco PIX password tool (4.0) #0: Tue Aug 22 23:22:19 PDT 2005
Flash=i28F640J5 @ 0x300
BIOS Flash=AT29C257 @ 0xd8000

Do you wish to erase the passwords? [yn] y
Passwords have been erased.

Rebooting....
```

Disabling Password Recovery

You might want to disable password recovery to ensure that unauthorized users cannot use the password recovery mechanism to compromise the security appliance. To disable password recovery, enter the following command:

```
hostname(config)# no service password-recovery
```

On the ASA 5500 series adaptive security appliance, the **no service password-recovery** command prevents a user from entering ROMMON mode with the configuration intact. When a user enters ROMMON mode, the security appliance prompts the user to erase all Flash file systems. The user cannot enter ROMMON mode without first performing this erasure. If a user chooses not to erase the Flash file system, the security appliance reloads. Because password recovery depends on using ROMMON mode and maintaining the existing configuration, this erasure prevents you from recovering a password. However, disabling password recovery prevents unauthorized users from viewing the configuration or inserting different passwords. In this case, to restore the system to an operating state, load a new image and a backup configuration file, if available.

The **service password-recovery** command appears in the configuration file for information only. When you enter the command at the CLI prompt, the setting is saved in NVRAM. The only way to change the setting is to enter the command at the CLI prompt. Loading a new configuration with a different version of the command does not change the setting. If you disable password recovery when the security appliance is configured to ignore the startup configuration at startup (in preparation for password recovery), then the security appliance changes the setting to load the startup configuration as usual. If you use failover, and the standby unit is configured to ignore the startup configuration, then the same change is made to the configuration register when the **no service password recovery** command replicates to the standby unit.

On the PIX 500 series security appliance, the **no service password-recovery** command forces the PIX password tool to prompt the user to erase all Flash file systems. The user cannot use the PIX password tool without first performing this erasure. If a user chooses not to erase the Flash file system, the security appliance reloads. Because password recovery depends on maintaining the existing configuration, this erasure prevents you from recovering a password. However, disabling password recovery prevents unauthorized users from viewing the configuration or inserting different passwords. In this case, to restore the system to an operating state, load a new image and a backup configuration file, if available.

Resetting the Password on the SSM Hardware Module

To reset the password to the default of "cisco" on the SSM hardware module, perform the following steps:

Step 1 Make sure that the SSM hardware module is in the Up state and supports password reset.

Step 2 Enter the following command:

```
hostname (config)# hw-module module 1 password-reset
```

Where *1* is the specified slot number on the SSM hardware module.

Note On the AIP SSM, entering this command reboots the hardware module. The module is offline until the rebooting is finished. Enter the **show module** command to monitor the module status. The AIP SSM supports this command in version 6.0 and later.

On the CSC SSM, entering this command resets web services on the hardware module after the password has been reset. You may lose connection to ASDM or be logged out of the hardware module. The CSC SSM supports this command in the most recent version of 6.1, dated November 2006.

```
Reset the password on module in slot 1? [confirm] y
```

Step 3 Enter **y** to confirm.

Using the ROM Monitor to Load a Software Image

This section describes how to load a software image to an adaptive security appliance from the ROM monitor mode using TFTP.

To load a software image to an adaptive security appliance, perform the following steps:

Step 1 Connect to the adaptive security appliance console port according to the instructions in "Accessing the Command-Line Interface" section on page 2-4.

Step 2 Power off the adaptive security appliance, and then power it on.

Step 3 During startup, press the **Escape** key when you are prompted to enter ROMMON mode.

Step 4 In ROMMOM mode, define the interface settings to the adaptive security appliance, including the IP address, TFTP server address, gateway address, software image file, and port, as follows:

```
rommon #1> ADDRESS=10.132.44.177
rommon #2> SERVER=10.129.0.30
rommon #3> GATEWAY=10.132.44.1
rommon #4> IMAGE=f1/asa800-232-k8.bin
rommon #5> PORT=Ethernet0/0
Ethernet0/0
Link is UP
MAC Address: 0012.d949.15b8
```

Note Be sure that the connection to the network already exists.

Step 5 To validate your settings, enter the **set** command.

```
rommon #6> set
ROMMON Variable Settings:
  ADDRESS=10.132.44.177
  SERVER=10.129.0.30
  GATEWAY=10.132.44.1
  PORT=Ethernet0/0
  VLAN=untagged
  IMAGE=f1/asa800-232-k8.bin
  CONFIG=
  LINKTIMEOUT=20
  PKTTIMEOUT=4
  RETRY=20
```

Step 6 Ping the TFTP server by entering the **ping server** command.

```
rommon #7> ping server
Sending 20, 100-byte ICMP Echoes to server 10.129.0.30, timeout is 4 seconds:

Success rate is 100 percent (20/20)
```

Step 7 Load the software image by entering the **tftp** command.

```
rommon #8> tftp
ROMMON Variable Settings:
  ADDRESS=10.132.44.177
  SERVER=10.129.0.30
  GATEWAY=10.132.44.1
  PORT=Ethernet0/0
  VLAN=untagged
  IMAGE=f1/asa800-232-k8.bin
  CONFIG=
  LINKTIMEOUT=20
  PKTTIMEOUT=4
  RETRY=20

tftp f1/asa800-232-k8.bin@10.129.0.30 via 10.132.44.1

Received 14450688 bytes

Launching TFTP Image...
Cisco PIX Security Appliance admin loader (3.0) #0: Mon Mar  5 16:00:07 MST 2007

Loading...
```

Step 8 Reload the security appliance to run the new image.

```
rommon #9> reload
```

After the software image is successfully loaded, the security appliance automatically exits ROMMOM mode.

Step 9 To verify that the correct software image has been loaded into the adaptive security appliance, check the version in the adaptive security appliance by entering the following command:

```
hostname> show version
```

Erasing the Flash File System

Step 1 Connect to the adaptive security appliance console port according to the instructions in "Accessing the Command-Line Interface" section on page 2-4.

Step 2 Power off the adaptive security appliance, and then power it on.

Step 3 During startup, press the **Escape** key when you are prompted to enter ROMMON mode.

Step 4 To erase the file system, enter the **erase** command, which overwrites all files and erases the file system, including hidden system files.

```
rommon #1> erase [disk0: | disk1: | flash:]
```

Other Troubleshooting Tools

The security appliance provides other troubleshooting tools that you can use. This section includes the following topics:

- Viewing Debug Messages, page 45-12
- Capturing Packets, page 45-12
- Viewing the Crash Dump, page 45-13

Viewing Debug Messages

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC. Moreover, it is best to use **debug** commands during periods of less network traffic and fewer users. Debugging during these periods decreases the likelihood that increased debug command processing overhead will affect system use. To enable debug messages, see the **debug** commands in the *Cisco Security Appliance Command Reference*.

Capturing Packets

Capturing packets is sometimes useful when troubleshooting connectivity problems or monitoring suspicious activity. We recommend contacting Cisco TAC if you want to use the packet capture feature. See the **capture** command in the *Cisco Security Appliance Command Reference*.

Viewing the Crash Dump

If the security appliance crashes, you can view the crash dump information. We recommend contacting Cisco TAC if you want to interpret the crash dump. See the **show crashdump** command in the *Cisco Security Appliance Command Reference*.

Common Problems

This section describes common problems with the security appliance, and how you might resolve them.

Symptom The context configuration was not saved, and was lost when you reloaded.

Possible Cause You did not save each context within the context execution space. If you are configuring contexts at the command line, you did not save the current context before you changed to the next context.

Recommended Action Save each context within the context execution space using the **copy run start** command. You cannot save contexts from the system execution space.

Symptom You cannot make a Telnet or SSH connection to the security appliance interface.

Possible Cause You did not enable Telnet or SSH to the security appliance.

Recommended Action Enable Telnet or SSH to the security appliance according to the instructions in "Allowing Telnet Access" section on page 42-1 or the "Allowing SSH Access" section on page 42-2.

Symptom You cannot ping the security appliance interface.

Possible Cause You disabled ICMP to the security appliance.

Recommended Action Enable ICMP to the security appliance for your IP address using the **icmp** command.

Symptom You cannot ping through the security appliance, although the access list allows it.

Possible Cause You did not enable the ICMP inspection engine or apply access lists on both the ingress and egress interfaces.

Recommended Action Because ICMP is a connectionless protocol, the security appliance does not automatically allow returning traffic through. In addition to an access list on the ingress interface, you either need to apply an access list to the egress interface to allow replying traffic, or enable the ICMP inspection engine, which treats ICMP connections as stateful connections.

Symptom Traffic does not pass between two interfaces on the same security level.

Possible Cause You did not enable the feature that allows traffic to pass between interfaces at the same security level.

Recommended Action Enable this feature according to the instructions in "Allowing Communication Between Interfaces on the Same Security Level" section on page 8-7.

Symptom IPSec tunnels do not duplicate during a failover to the standby device.

Possible Cause The switch port that the ASA is plugged into is set to 10/100 instead of 1000.

Recommended Action Set the switch port that the ASA is plugged into to 1000.



PART 5

Reference



APPENDIX A

Sample Configurations

This appendix illustrates and describes a number of common ways to implement the security appliance, and includes the following sections:

- Example 1: Multiple Mode Firewall With Outside Access, page A-1
- Example 2: Single Mode Firewall Using Same Security Level, page A-6
- Example 3: Shared Resources for Multiple Contexts, page A-8
- Example 4: Multiple Mode, Transparent Firewall with Outside Access, page A-13
- Example 5: Single Mode, Transparent Firewall with NAT, page A-18
- Example 6: IPv6 Configuration, page A-19
- Example 7: Dual ISP Support Using Static Route Tracking, page A-20
- Example 8: Multicast Routing, page A-21
- Example 9: LAN-Based Active/Standby Failover (Routed Mode), page A-24
- Example 10: LAN-Based Active/Active Failover (Routed Mode), page A-25
- Example 11: LAN-Based Active/Standby Failover (Transparent Mode), page A-28
- Example 12: LAN-Based Active/Active Failover (Transparent Mode), page A-30
- Example 13: Cable-Based Active/Standby Failover (Routed Mode), page A-34
- Example 14: Cable-Based Active/Standby Failover (Transparent Mode), page A-35
- Example 15: ASA 5505 Base License, page A-36
- Example 16: ASA 5505 Security Plus License with Failover and Dual-ISP Backup, page A-38
- Example 17: AIP SSM in Multiple Context Mode, page A-40

Example 1: Multiple Mode Firewall With Outside Access

This configuration creates three security contexts plus the admin context, each with an inside and an outside interface. Both interfaces are configured as redundant interfaces.

The Customer C context includes a DMZ interface where a Websense server for HTTP filtering resides on the service provider premises (see Figure A-1).

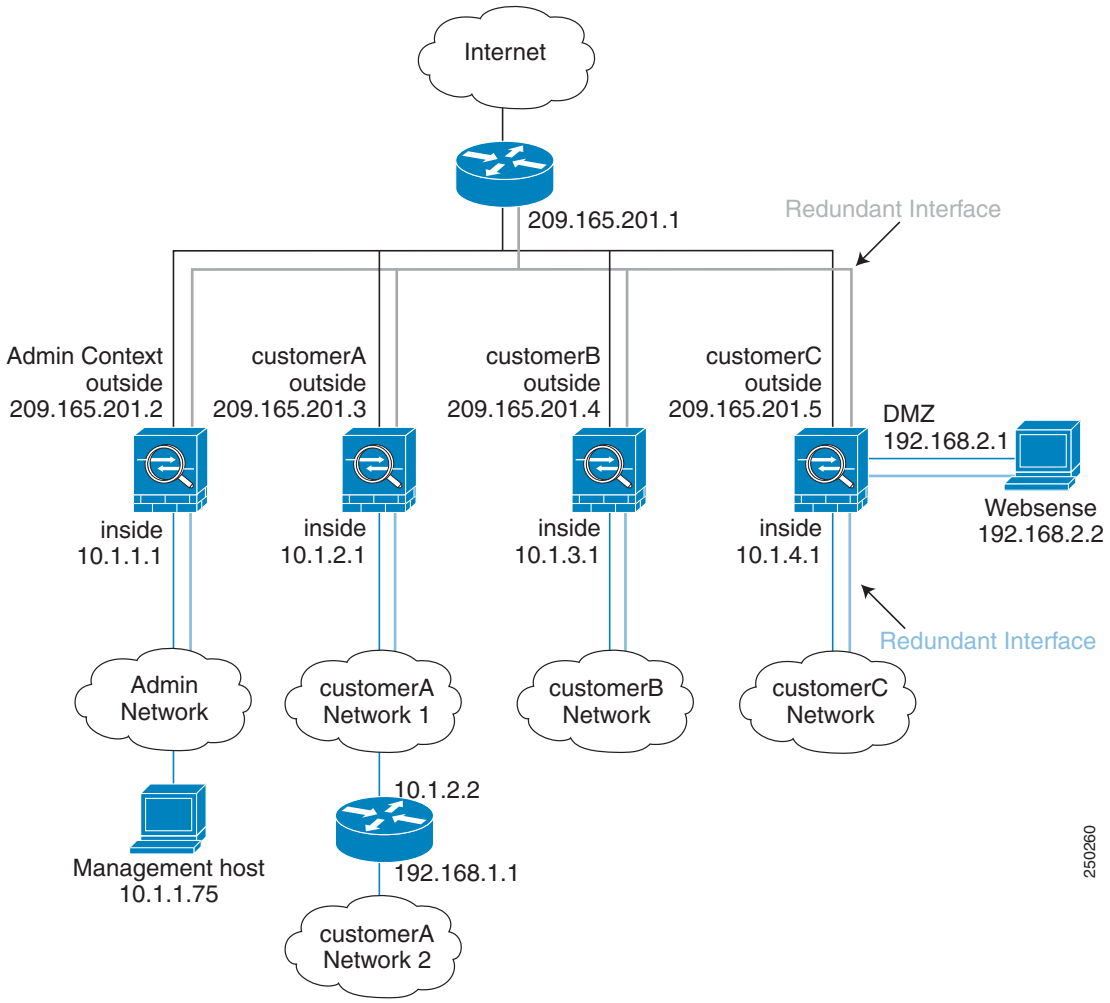
Inside hosts can access the Internet through the outside using dynamic NAT or PAT, but no outside hosts can access the inside.

The Customer A context has a second network behind an inside router.

The admin context allows SSH sessions to the security appliance from one host.

Note Although inside IP addresses can be the same across contexts when the interfaces are unique, keeping them unique is easier to manage.

Figure A-1 Example 1



See the following sections for the configurations for this scenario:

- System Configuration for Example 1, page A-3
- Admin Context Configuration for Example 1, page A-4
- Customer A Context Configuration for Example 1, page A-4
- Customer B Context Configuration for Example 1, page A-5
- Customer C Context Configuration for Example 1, page A-5

System Configuration for Example 1

You must first enable multiple context mode using the **mode multiple** command. The mode is not stored in the configuration file, even though it endures reboots. Enter the **show mode** command to view the current mode.

```
hostname Farscape
password passw0rd
enable password chr1cht0n
mac-address auto
asdm image disk0:/asdm.bin
boot system disk0:/image.bin
admin-context admin
interface gigabitethernet 0/0
   no shutdown
interface gigabitethernet 0/1
   no shutdown
interface gigabitethernet 0/2
   no shutdown
interface gigabitethernet 0/3
   no shutdown
interface redundant 1
   member-interface gigabitethernet 0/0
   member-interface gigabitethernet 0/1
interface redundant 2
   member-interface gigabitethernet 0/2
   member-interface gigabitethernet 0/3
interface redundant 1.3
   vlan 3
   no shutdown
interface redundant 2.4
   vlan 4
   no shutdown
interface redundant 2.5
   vlan 5
   no shutdown
interface redundant 2.6
   vlan 6
   no shutdown
interface redundant 2.7
   vlan 7
   no shutdown
interface redundant 2.8
   vlan 8
   no shutdown
class gold
   limit-resource rate conns 2000
   limit-resource conns 20000
class silver
   limit-resource rate conns 1000
   limit-resource conns 10000
class bronze
   limit-resource rate conns 500
   limit-resource conns 5000
context admin
   allocate-interface redundant1.3 int1
   allocate-interface redundant2.4 int2
   config-url disk0://admin.cfg
   member default
context customerA
   description This is the context for customer A
   allocate-interface redundant1.3 int1
   allocate-interface redundant2.5 int2
```

```
   config-url disk0://contexta.cfg
   member gold
context customerB
   description This is the context for customer B
   allocate-interface redundant1.3 int1
   allocate-interface redundant2.6 int2
   config-url disk0://contextb.cfg
   member silver
context customerC
   description This is the context for customer C
   allocate-interface redundant1.3 int1
   allocate-interface redundant2.7-redundant2.8 int2-int3
   config-url disk0://contextc.cfg
   member bronze
```

Admin Context Configuration for Example 1

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

The host at 10.1.1.75 can access the context using SSH, which requires a key to be generated using the **crypto key generate** command.

```
hostname Admin
domain example.com
interface int1
   nameif outside
   security-level 0
   ip address 209.165.201.2 255.255.255.224
   no shutdown
interface int2
   nameif inside
   security-level 100
   ip address 10.1.1.1 255.255.255.0
   no shutdown
passwd secret1969
enable password h1andl0
route outside 0 0 209.165.201.1 1
ssh 10.1.1.75 255.255.255.255 inside
nat (inside) 1 10.1.1.0 255.255.255.0
! This context uses dynamic NAT for inside users that access the outside
global (outside) 1 209.165.201.10-209.165.201.29
! The host at 10.1.1.75 has access to the Websense server in Customer C, so
! it needs a static translation for use in Customer C's access list
static (inside,outside) 209.165.201.30 10.1.1.75 netmask 255.255.255.255
```

Customer A Context Configuration for Example 1

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
interface int1
   nameif outside
   security-level 0
   ip address 209.165.201.3 255.255.255.224
   no shutdown
interface int2
   nameif inside
```

```
   security-level 100
   ip address 10.1.2.1 255.255.255.0
   no shutdown
passwd hell0!
enable password enter55
route outside 0 0 209.165.201.1 1
! The Customer A context has a second network behind an inside router that requires a
! static route. All other traffic is handled by the default route pointing to the router.
route inside 192.168.1.0 255.255.255.0 10.1.2.2 1
nat (inside) 1 10.1.2.0 255.255.255.0
! This context uses dynamic PAT for inside users that access that outside. The outside
! interface address is used for the PAT address
global (outside) 1 interface
```

Customer B Context Configuration for Example 1

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
interface int1
   nameif outside
   security-level 0
   ip address 209.165.201.4 255.255.255.224
   no shutdown
interface int2
   nameif inside
   security-level 100
   ip address 10.1.3.1 255.255.255.0
   no shutdown
passwd tenac10us
enable password defen$e
route outside 0 0 209.165.201.1 1
nat (inside) 1 10.1.3.0 255.255.255.0
! This context uses dynamic PAT for inside users that access the outside
global (outside) 1 209.165.201.9 netmask 255.255.255.255
access-list INTERNET remark Inside users only access HTTP and HTTPS servers on the outside
access-list INTERNET extended permit tcp any any eq http
access-list INTERNET extended permit tcp any any eq https
access-group INTERNET in interface inside
```

Customer C Context Configuration for Example 1

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
interface int1
   nameif outside
   security-level 0
   ip address 209.165.201.5 255.255.255.224
   no shutdown
interface int2
   nameif inside
   security-level 100
   ip address 10.1.4.1 255.255.255.0
   no shutdown
interface int3
   nameif dmz
   security-level 50
```

```
   ip address 192.168.2.1 255.255.255.0
   no shutdown
passwd fl0wer
enable password treeh0u$e
route outside 0 0 209.165.201.1 1
url-server (dmz) vendor websense host 192.168.2.2 url-block block 50
url-cache dst 128
filter url http 10.1.4.0 255.255.255.0 0 0
! When inside users access an HTTP server, the security appliance consults with a
! Websense server to determine if the traffic is allowed
nat (inside) 1 10.1.4.0 255.255.255.0
! This context uses dynamic NAT for inside users that access the outside
global (outside) 1 209.165.201.9 netmask 255.255.255.255
! A host on the admin context requires access to the Websense server for management using
! pcAnywhere, so the Websense server uses a static translation for its private address
static (dmz,outside) 209.165.201.6 192.168.2.2 netmask 255.255.255.255
access-list MANAGE remark Allows the management host to use pcAnywhere on the Websense
server
access-list MANAGE extended permit tcp host 209.165.201.30 host 209.165.201.6 eq
pcanywhere-data
access-list MANAGE extended permit udp host 209.165.201.30 host 209.165.201.6 eq
pcanywhere-status
access-group MANAGE in interface outside
```

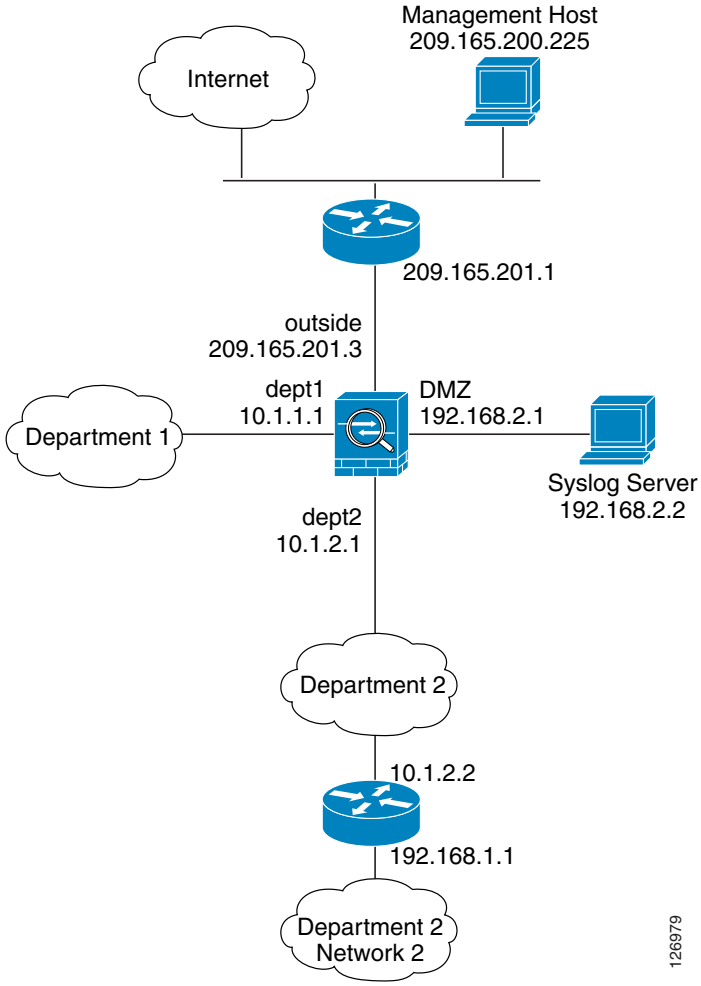
Example 2: Single Mode Firewall Using Same Security Level

This configuration creates three internal interfaces. Two of the interfaces connect to departments that are on the same security level, which allows all hosts to communicate without using access lists. The DMZ interface hosts a syslog server. The management host on the outside needs access to the Syslog server and the security appliance. The security appliance uses RIP on the inside interfaces to learn routes. The security appliance does not advertise routes with RIP; the upstream router needs to use static routes for security appliance traffic (see Figure A-2).

The Department networks are allowed to access the Internet, and use PAT.

Threat detection is enabled.

Figure A-2 Example 2




```
passwd g00fba11
enable password gen1u$
hostname Buster
asdm image disk0:/asdm.bin
boot system disk0:/image.bin
interface gigabitethernet 0/0
   nameif outside
   security-level 0
   ip address 209.165.201.3 255.255.255.224
   no shutdown
interface gigabitethernet 0/1
   nameif dept2
   security-level 100
   ip address 10.1.2.1 255.255.255.0
   mac-address 000C.F142.4CDE standby 000C.F142.4CDF
   no shutdown
   rip authentication mode md5
   rip authentication key scorpius key_id 1
interface gigabitethernet 0/2
   nameif dept1
   security-level 100
   ip address 10.1.1.1 255.255.255.0
   no shutdown
```

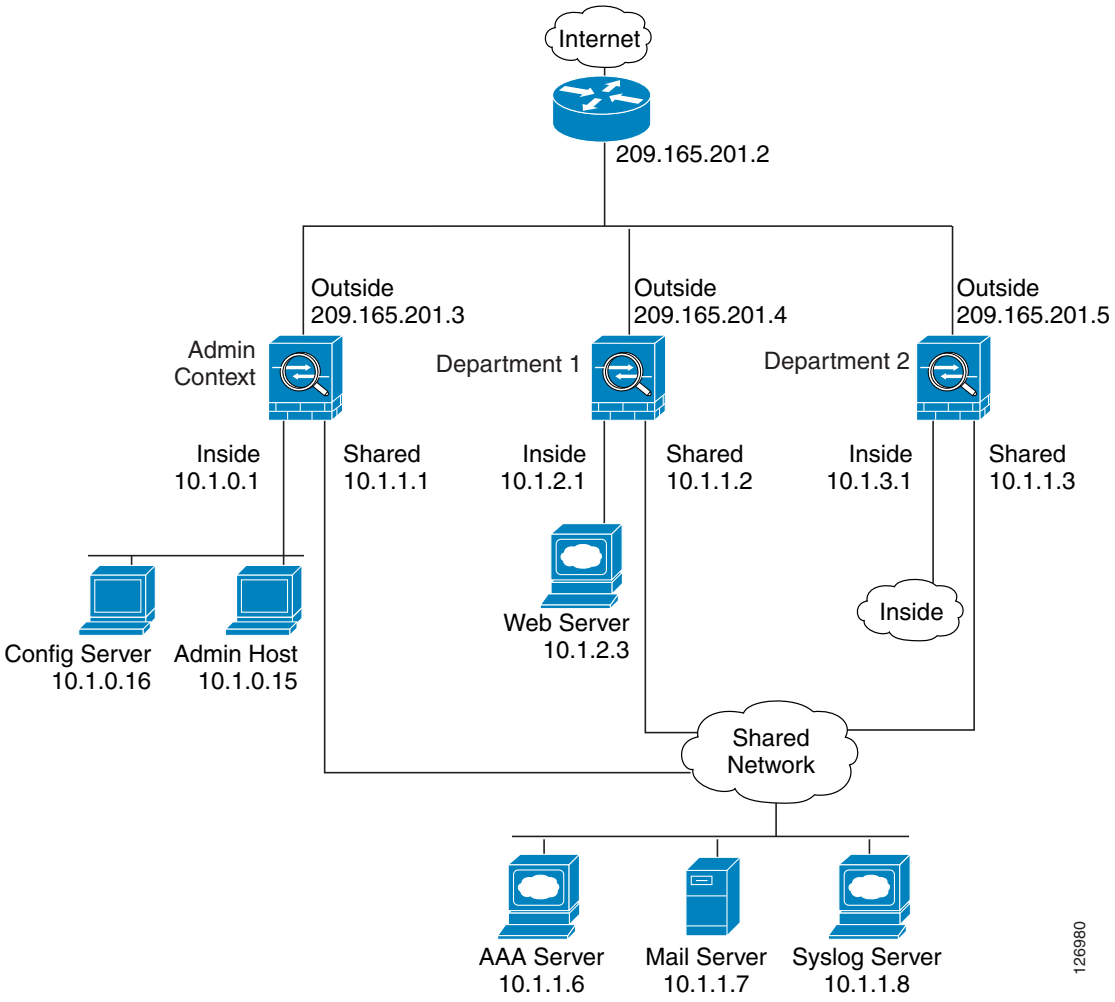
```
interface gigabitethernet 0/3
   nameif dmz
   security-level 50
   ip address 192.168.2.1 255.255.255.0
   no shutdown
same-security-traffic permit inter-interface
route outside 0 0 209.165.201.1 1
nat (dept1) 1 10.1.1.0 255.255.255.0
nat (dept2) 1 10.1.2.0 255.255.255.0
! The dept1 and dept2 networks use PAT when accessing the outside
global (outside) 1 209.165.201.9 netmask 255.255.255.255
! Because we perform dynamic NAT on these addresses for outside access, we need to perform
! NAT on them for all other interface access. This identity static statement just
! translates the local address to the same address.
static (dept1,dept2) 10.1.1.0 10.1.1.0 netmask 255.255.255.0
static (dept2,dept1) 10.1.2.0 10.1.2.0 netmask 255.255.255.0
! The syslog server uses a static translation so the outside management host can access
! the server
static (dmz,outside) 209.165.201.5 192.168.2.2 netmask 255.255.255.255
access-list MANAGE remark Allows the management host to access the syslog server
access-list MANAGE extended permit tcp host 209.165.200.225 host 209.165.201.5 eq ssh
access-group MANAGE in interface outside
! Advertises the security appliance IP address as the default gateway for the downstream
! router. The security appliance does not advertise a default route to the upstream
! router. Listens for RIP updates from the downstream router. The security appliance does
! not listen for RIP updates from the upstream router because a default route to the
! upstream router is all that is required.
router rip
   network 10.0.0.0
   default information originate
   version 2
ssh 209.165.200.225 255.255.255.255 outside
logging trap 5
! System messages are sent to the syslog server on the DMZ network
logging host dmz 192.168.2.2
logging enable
! Enable basic threat detection:
threat-detection basic-threat
threat-detection rate dos-drop rate-interval 600 average-rate 60 burst-rate 100
! Enables scanning threat detection and automatically shun attackers,
! except for hosts on the 10.1.1.0 network:
threat-detection scanning-threat shun except ip-address 10.1.1.0 255.255.255.0
threat-detection rate scanning-threat rate-interval 1200 average-rate 10 burst-rate 20
threat-detection rate scanning-threat rate-interval 2400 average-rate 10 burst-rate 20
! Enable statistics for access-lists:
threat-detection statistics access-list
```

Example 3: Shared Resources for Multiple Contexts

This configuration includes multiple contexts for multiple departments within a company. Each department has its own security context so that each department can have its own security policy. However, the syslog, mail, and AAA servers are shared across all departments. These servers are placed on a shared interface (see Figure A-3).

Department 1 has a web server that outside users who are authenticated by the AAA server can access.

Figure A-3 Example 3



See the following sections for the configurations for this scenario:

- System Configuration for Example 3, page A-9
- Admin Context Configuration for Example 3, page A-10
- Department 1 Context Configuration for Example 3, page A-11
- Department 2 Context Configuration for Example 3, page A-12

System Configuration for Example 3

You must first enable multiple context mode using the **mode multiple** command. The mode is not stored in the configuration file, even though it endures reboots. Enter the **show mode** command to view the current mode.

```
hostname Ubik
password pkd55
enable password deckard69
asdm image disk0:/asdm.bin
boot system disk0:/image.bin
```

```
mac-address auto
admin-context admin
interface gigabitethernet 0/0
   no shutdown
interface gigabitethernet 0/0.200
! This is the shared outside interface
   vlan 200
   no shutdown
interface gigabitethernet 0/1
   no shutdown
interface gigabitethernet 0/1.201
! This is the inside interface for admin
   vlan 201
   no shutdown
interface gigabitethernet 0/1.202
! This is the inside interface for department 1
   vlan 202
   no shutdown
interface gigabitethernet 0/1.203
! This is the inside interface for department 2
   vlan 203
   no shutdown
interface gigabitethernet 0/1.300
! This is the shared inside interface
   vlan 300
   no shutdown
context admin
   allocate-interface gigabitethernet 0/0.200
   allocate-interface gigabitethernet 0/1.201
   allocate-interface gigabitethernet 0/1.300
   config-url disk0://admin.cfg
context department1
   allocate-interface gigabitethernet 0/0.200
   allocate-interface gigabitethernet 0/1.202
   allocate-interface gigabitethernet 0/1.300
   config-url ftp://admin:passw0rd@10.1.0.16/dept1.cfg
context department2
   allocate-interface gigabitethernet 0/0.200
   allocate-interface gigabitethernet 0/1.203
   allocate-interface gigabitethernet 0/1.300
   config-url ftp://admin:passw0rd@10.1.0.16/dept2.cfg
```

Admin Context Configuration for Example 3

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
hostname Admin
interface gigabitethernet 0/0.200
   nameif outside
   security-level 0
   ip address 209.165.201.3 255.255.255.224
   no shutdown
interface gigabitethernet 0/0.201
   nameif inside
   security-level 100
   ip address 10.1.0.1 255.255.255.0
   no shutdown
interface gigabitethernet 0/0.300
   nameif shared
   security-level 50
```

```
   ip address 10.1.1.1 255.255.255.0
   no shutdown
passwd v00d00
enable password d011
route outside 0 0 209.165.201.2 1
nat (inside) 1 10.1.0.0 255.255.255.0
! This context uses PAT for inside users that access the outside
global (outside) 1 209.165.201.6 netmask 255.255.255.255
! This context uses PAT for inside users that access the shared network
global (shared) 1 10.1.1.30
! Because this host can access the web server in the Department 1 context, it requires a
! static translation
static (inside,outside) 209.165.201.7 10.1.0.15 netmask 255.255.255.255
! Because this host has management access to the servers on the Shared interface, it
! requires a static translation to be used in an access list
static (inside,shared) 10.1.1.78 10.1.0.15 netmask 255.255.255.255
access-list SHARED remark -Allows only mail traffic from inside to exit shared interface
access-list SHARED remark -but allows the admin host to access any server.
access-list SHARED extended permit ip host 10.1.1.78 any
access-list SHARED extended permit tcp host 10.1.1.30 host 10.1.1.7 eq smtp
! Note that the translated addresses are used.
access-group SHARED out interface shared
! Allows 10.1.0.15 to access the admin context using Telnet. From the admin context, you
! can access all other contexts.
telnet 10.1.0.15 255.255.255.255 inside
aaa-server AAA-SERVER protocol tacacs+
aaa-server AAA-SERVER (shared) host 10.1.1.6
   key TheUauthKey
   server-port 16
! The host at 10.1.0.15 must authenticate with the AAA server to log in
aaa authentication telnet console AAA-SERVER
aaa authorization command AAA-SERVER LOCAL
aaa accounting command AAA-SERVER
logging trap 6
! System messages are sent to the syslog server on the Shared network
logging host shared 10.1.1.8
logging enable
```

Department 1 Context Configuration for Example 3

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
interface gigabitethernet 0/0.200
   nameif outside
   security-level 0
   ip address 209.165.201.4 255.255.255.224
   no shutdown
interface gigabitethernet 0/0.202
   nameif inside
   security-level 100
   ip address 10.1.2.1 255.255.255.0
   no shutdown
interface gigabitethernet 0/0.300
   nameif shared
   security-level 50
   ip address 10.1.1.2 255.255.255.0
   no shutdown
passwd cugel
enable password rhialto
nat (inside) 1 10.1.2.0 255.255.255.0
```

```
! The inside network uses PAT when accessing the outside
global (outside) 1 209.165.201.8 netmask 255.255.255.255
! The inside network uses dynamic NAT when accessing the shared network
global (shared) 1 10.1.1.31-10.1.1.37
! The web server can be accessed from outside and requires a static translation
static (inside,outside) 209.165.201.9 10.1.2.3 netmask 255.255.255.255
access-list WEBSERVER remark -Allows the management host (its translated address) on the
access-list WEBSERVER remark -admin context to access the web server for management
access-list WEBSERVER remark -it can use any IP protocol
access-list WEBSERVER extended permit ip host 209.165.201.7 host 209.165.201.9
access-list WEBSERVER remark -Allows any outside address to access the web server
access-list WEBSERVER extended permit tcp any eq http host 209.165.201.9 eq http
access-group WEBSERVER in interface outside
access-list MAIL remark -Allows only mail traffic from inside to exit out the shared int
! Note that the translated addresses are used.
access-list MAIL extended permit tcp host 10.1.1.31 eq smtp host 10.1.1.7 eq smtp
access-list MAIL extended permit tcp host 10.1.1.32 eq smtp host 10.1.1.7 eq smtp
access-list MAIL extended permit tcp host 10.1.1.33 eq smtp host 10.1.1.7 eq smtp
access-list MAIL extended permit tcp host 10.1.1.34 eq smtp host 10.1.1.7 eq smtp
access-list MAIL extended permit tcp host 10.1.1.35 eq smtp host 10.1.1.7 eq smtp
access-list MAIL extended permit tcp host 10.1.1.36 eq smtp host 10.1.1.7 eq smtp
access-list MAIL extended permit tcp host 10.1.1.37 eq smtp host 10.1.1.7 eq smtp
access-group MAIL out interface shared
aaa-server AAA-SERVER protocol tacacs+
aaa-server AAA-SERVER (shared) host 10.1.1.6
   key TheUauthKey
   server-port 16
! All traffic matching the WEBSERVER access list must authenticate with the AAA server
aaa authentication match WEBSERVER outside AAA-SERVER
logging trap 4
! System messages are sent to the syslog server on the Shared network
logging host shared 10.1.1.8
logging enable
```

Department 2 Context Configuration for Example 3

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
interface gigabitethernet 0/0.200
   nameif outside
   security-level 0
   ip address 209.165.201.5 255.255.255.224
   no shutdown
interface gigabitethernet 0/0.203
   nameif inside
   security-level 100
   ip address 10.1.3.1 255.255.255.0
   no shutdown
interface gigabitethernet 0/0.300
   nameif shared
   security-level 50
   ip address 10.1.1.3 255.255.255.0
   no shutdown
passwd maz1r1an
enable password ly0ne$$e
route outside 0 0 209.165.201.2 1
nat (inside) 1 10.1.3.0 255.255.255.0
! The inside network uses PAT when accessing the outside
global (outside) 1 209.165.201.10 netmask 255.255.255.255
! The inside network uses PAT when accessing the shared network
```

```
global (shared) 1 10.1.1.38
access-list MAIL remark -Allows only mail traffic from inside to exit out the shared int
access-list MAIL extended permit tcp host 10.1.1.38 host 10.1.1.7 eq smtp
! Note that the translated PAT address is used.
access-group MAIL out interface shared
logging trap 3
! System messages are sent to the syslog server on the Shared network
logging host shared 10.1.1.8
logging enable
```

Example 4: Multiple Mode, Transparent Firewall with Outside Access

This configuration creates three security contexts plus the admin context. Each context allows OSPF traffic to pass between the inside and outside routers (see Figure A-4).

Inside hosts can access the Internet through the outside, but no outside hosts can access the inside.

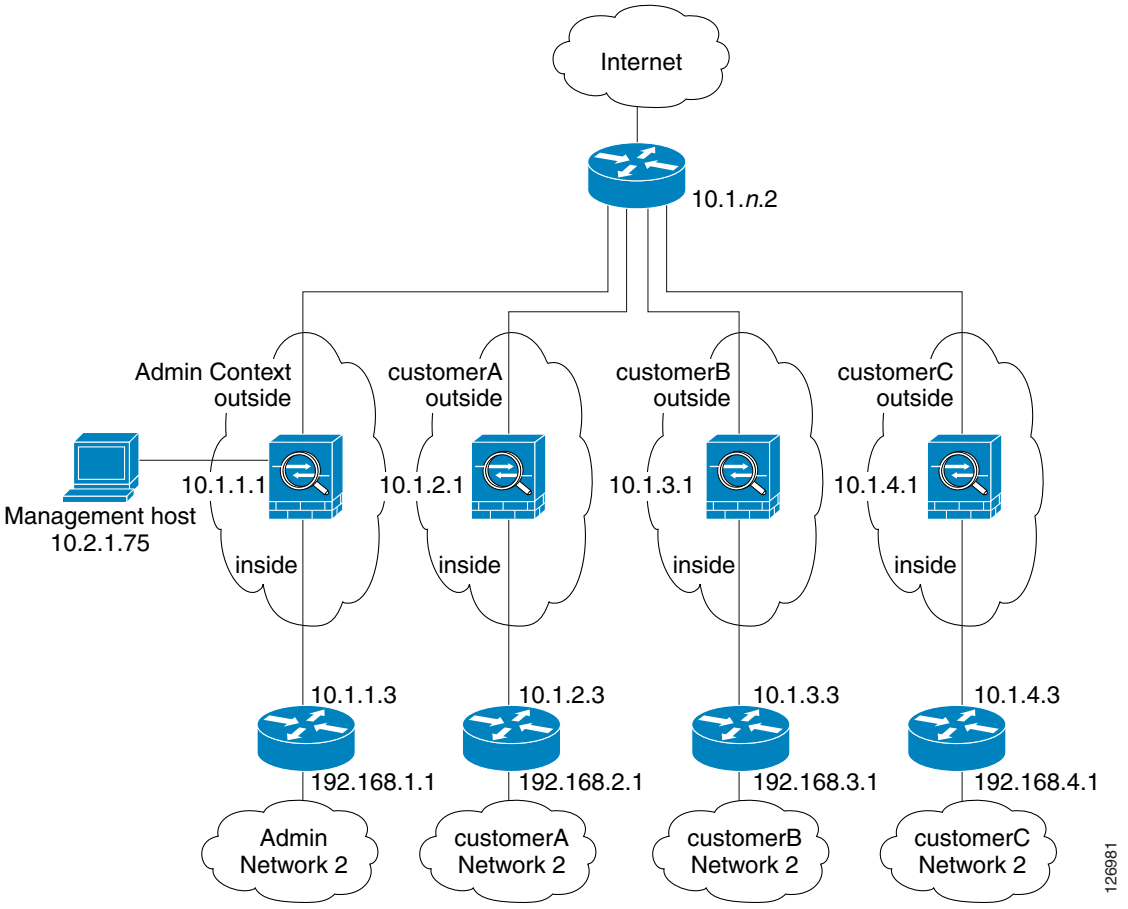
An out-of-band management host is connected to the Management 0/0 interface.

The admin context allows SSH sessions to the security appliance from one host.

Connection limit settings for each context, except admin, limit the number of connections to guard against DoS attacks.

Note Although inside IP addresses can be the same across contexts, keeping them unique is easier to manage.

Figure A-4 Example 4



See the following sections for the configurations for this scenario:

- System Configuration for Example 4, page A-14
- Admin Context Configuration for Example 4, page A-15
- Customer A Context Configuration for Example 4, page A-16
- Customer B Context Configuration for Example 4, page A-16
- Customer C Context Configuration for Example 4, page A-17

System Configuration for Example 4

You must first enable multiple context mode using the **mode multiple** command. The mode is not stored in the configuration file, even though it endures reboots. Enter the **show mode** command to view the current mode.

```
firewall transparent
hostname Farscape
password passw0rd
enable password chr1cht0n
asdm image disk0:/asdm.bin
boot system disk0:/image.bin
mac-address auto
admin-context admin
```

```
interface gigabitethernet 0/0
   no shutdown
interface gigabitethernet 0/0.150
   vlan 150
   no shutdown
interface gigabitethernet 0/0.151
   vlan 151
   no shutdown
interface gigabitethernet 0/0.152
   vlan 152
   no shutdown
interface gigabitethernet 0/0.153
   vlan 153
   no shutdown
interface gigabitethernet 0/1
   shutdown
interface gigabitethernet 0/1.4
   vlan 4
   no shutdown
interface gigabitethernet 0/1.5
   vlan 5
   no shutdown
interface gigabitethernet 0/1.6
   vlan 6
   no shutdown
interface gigabitethernet 0/1.7
   vlan 7
   no shutdown
interface management 0/0
   no shutdown
context admin
   allocate-interface gigabitethernet 0/0.150
   allocate-interface gigabitethernet 0/1.4
   allocate-interface management 0/0
   config-url disk0://admin.cfg
context customerA
   description This is the context for customer A
   allocate-interface gigabitethernet 0/0.151
   allocate-interface gigabitethernet 0/1.5
   config-url disk0://contexta.cfg
context customerB
   description This is the context for customer B
   allocate-interface gigabitethernet 0/0.152
   allocate-interface gigabitethernet 0/1.6
   config-url disk0://contextb.cfg
context customerC
   description This is the context for customer C
   allocate-interface gigabitethernet 0/0.153
   allocate-interface gigabitethernet 0/1.7
   config-url disk0://contextc.cfg
```

Admin Context Configuration for Example 4

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

The host at 10.2.1.75 can access the context using SSH, which requires a key pair to be generated using the **crypto key generate** command.

```
hostname Admin
domain example.com
```

```
interface gigabitethernet 0/0.150
   nameif outside
   security-level 0
   no shutdown
interface gigabitethernet 0/1.4
   nameif inside
   security-level 100
   no shutdown
interface management 0/0
   nameif manage
   security-level 50
! Unlike other transparent interfaces, the management interface
! requires an IP address:
   ip address 10.2.1.1 255.255.255.0
   no shutdown
passwd secret1969
enable password h1andl0
ip address 10.1.1.1 255.255.255.0
route outside 0 0 10.1.1.2 1
ssh 10.2.1.75 255.255.255.255 manage
access-list OSPF remark -Allows OSPF
access-list OSPF extended permit 89 any any
access-group OSPF in interface outside
```

Customer A Context Configuration for Example 4

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
interface gigabitethernet 0/0.151
   nameif outside
   security-level 0
   no shutdown
interface gigabitethernet 0/1.5
   nameif inside
   security-level 100
   no shutdown
passwd hell0!
enable password enter55
ip address 10.1.2.1 255.255.255.0
route outside 0 0 10.1.2.2 1
access-list OSPF remark -Allows OSPF
access-list OSPF extended permit 89 any any
access-group OSPF in interface outside
```

Customer B Context Configuration for Example 4

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
interface gigabitethernet 0/0.152
   nameif outside
   security-level 0
   no shutdown
interface gigabitethernet 0/1.6
   nameif inside
   security-level 100
   no shutdown
```

```
passwd tenac10us
enable password defen$e
ip address 10.1.3.1 255.255.255.0
route outside 0 0 10.1.3.2 1
access-list OSPF remark -Allows OSPF
access-list OSPF extended permit 89 any any
access-group OSPF in interface outside
! The following commands add connection limits to the global policy.
class-map conn_limits
 match any
policy-map global_policy
   class conn_limits
      set connection conn-max 5000 embryonic-conn-max 2000
      set connection timeout tcp 2:0:0 reset half-close 0:5:0 embryonic 0:0:20 dcd 20 3
service-policy global_policy global
```

Customer C Context Configuration for Example 4

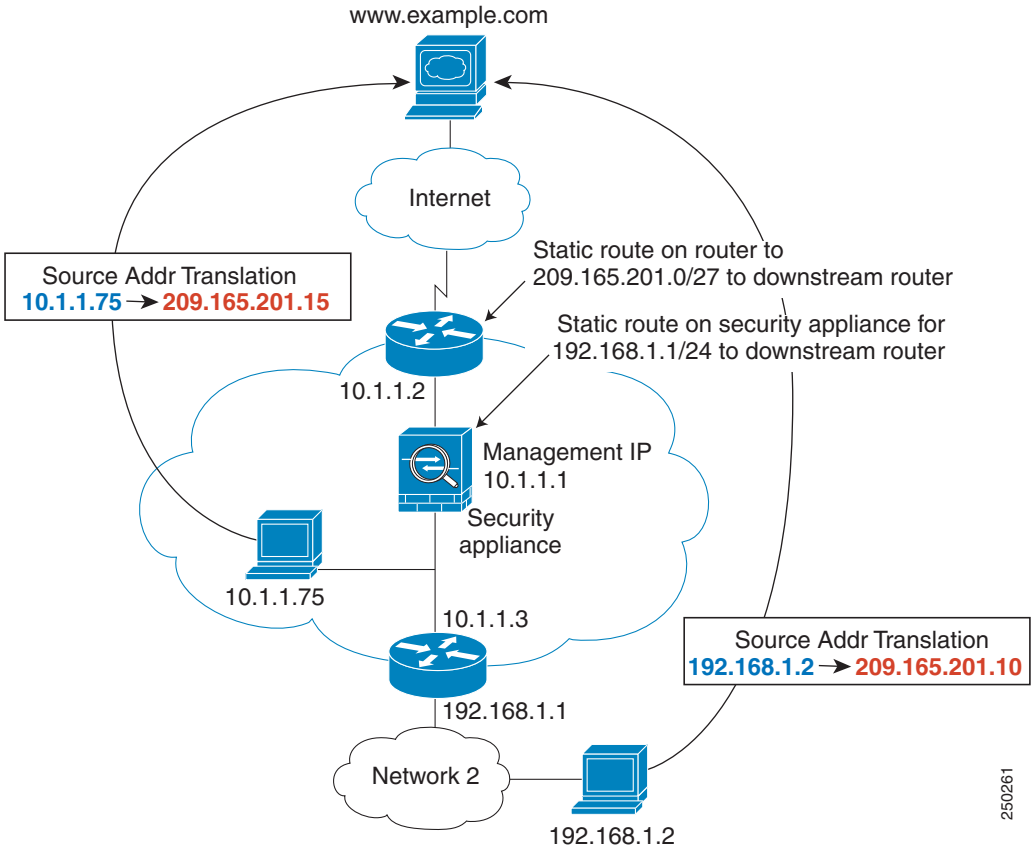
To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
interface gigabitethernet 0/0.153
   nameif outside
   security-level 0
   no shutdown
interface gigabitethernet 0/1.7
   nameif inside
   security-level 100
   no shutdown
passwd fl0wer
enable password treeh0u$e
ip address 10.1.4.1 255.255.255.0
route outside 0 0 10.1.4.2 1
access-list OSPF remark -Allows OSPF
access-list OSPF extended permit 89 any any
access-group OSPF in interface outside
! The following commands add connection limits to the global policy.
class-map conn_limits
 match any
policy-map global_policy
   class conn_limits
      set connection conn-max 5000 embryonic-conn-max 2000
      set connection timeout tcp 2:0:0 reset half-close 0:5:0 embryonic 0:0:20 dcd 20 3
service-policy global_policy global
```

Example 5: Single Mode, Transparent Firewall with NAT

This configuration shows how to configure NAT in transparent mode (see Figure A-5).

Figure A-5 Example 5



The host at 10.1.1.75 can access the security appliance using SSH, which requires a key pair to be generated using the **crypto key generate** command.

```
firewall transparent
hostname Farscape
password passw0rd
enable password chr1cht0n
asdm image disk0:/asdm.bin
boot system disk0:/image.bin
hostname Moya
domain example.com
interface gigabitethernet 0/0
   nameif outside
   security-level 0
   no shutdown
interface gigabitethernet 0/1
   nameif inside
   security-level 100
   no shutdown
ip address 10.1.1.1 255.255.255.0
route outside 0 0 10.1.1.2 1
! The following route is required when you perform NAT
```

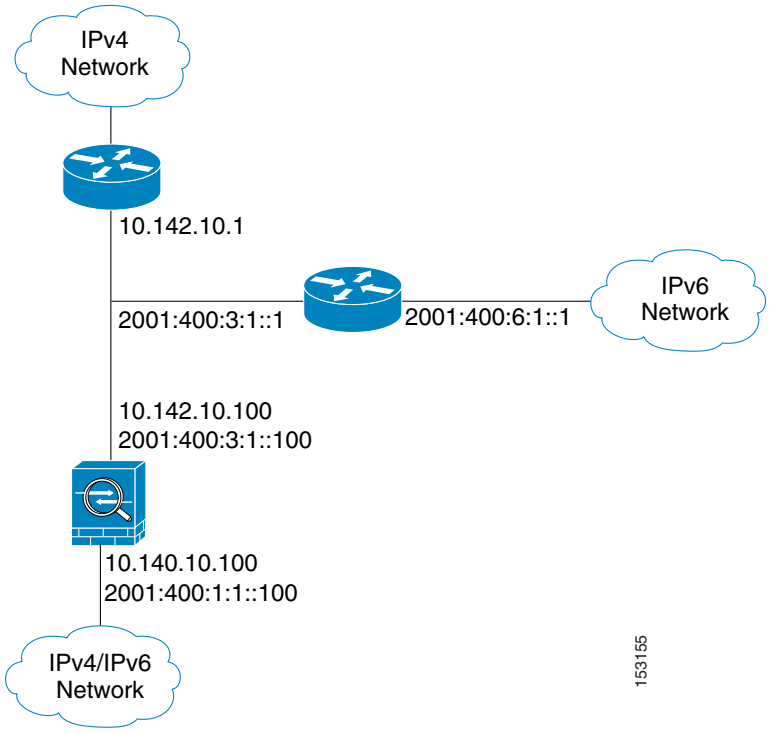
```
! on non-directly-connected networks:
route inside 192.168.1.0 255.255.255.0 10.1.1.3 1
ssh 10.1.1.75 255.255.255.255 inside
nat (inside) 1 10.1.1.0 255.255.255.0
nat (inside) 1 198.168.1.0 255.255.255.0
global (outside) 1 209.165.201.1-209.165.201.15
```

Example 6: IPv6 Configuration

This sample configuration shows several features of IPv6 support on the security appliance:

- Each interface is configured with both IPv6 and IPv4 addresses.
- The IPv6 default route is set with the **ipv6 route** command.
- An IPv6 access list is applied to the outside interface.
- The enforcement of Modified-EUI64 format interface identifiers in the IPv6 addresses of hosts on the inside interface.
- The outside interface suppresses router advertisement messages.
- An IPv6 static route.

Figure A-6 IPv6 Dual Stack Configuration



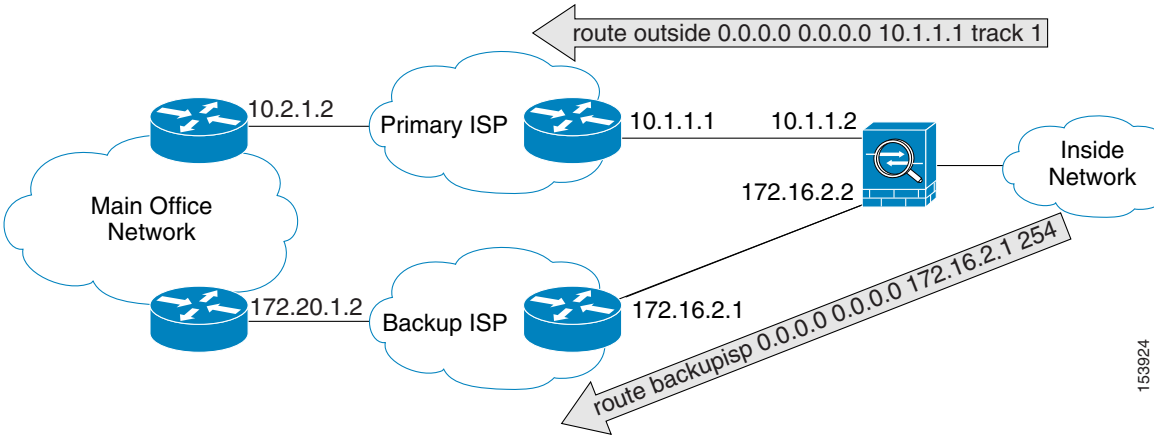

```
enable password myenablepassword
passwd mypassword
hostname coyupix
asdm image flash:/asdm.bin
boot system flash:/image.bin
```

```
interface gigabitethernet0/0
   nameif outside
   security-level 0
   ip address 10.142.10.100 255.255.255.0
   ipv6 address 2001:400:3:1::100/64
   ipv6 nd suppress-ra
   ospf mtu-ignore auto
   no shutdown
interface gigabitethernet0/1
   nameif inside
   security-level 100
   ip address 10.140.10.100 255.255.255.0
   ipv6 address 2001:400:1:1::100/64
   ospf mtu-ignore auto
   no shutdown
access-list allow extended permit icmp any any
ssh 10.140.10.75 255.255.255.255 inside
logging enable
logging buffered debugging
ipv6 enforce-eui64 inside
ipv6 route outside 2001:400:6:1::/64 2001:400:3:1::1
ipv6 route outside ::/0 2001:400:3:1::1
ipv6 access-list outacl permit icmp6 2001:400:2:1::/64 2001:400:1:1::/64
ipv6 access-list outacl permit tcp 2001:400:2:1::/64 2001:400:1:1::/64 eq telnet
ipv6 access-list outacl permit tcp 2001:400:2:1::/64 2001:400:1:1::/64 eq ftp
ipv6 access-list outacl permit tcp 2001:400:2:1::/64 2001:400:1:1::/64 eq www
access-group allow in interface outside
access-group outacl in interface outside
route outside 0.0.0.0 0.0.0.0 16.142.10.1 1
```

Example 7: Dual ISP Support Using Static Route Tracking

This configuration shows a remote office using static route tracking to use a backup ISP route if the primary ISP route fails. The security appliance in the remote office uses ICMP echo requests to monitor the availability of the main office gateway. If that gateway becomes unavailable through the default route, the default route is removed from the routing table and the floating route to the backup ISP is used in its place.

Figure A-7 Dual ISP Support

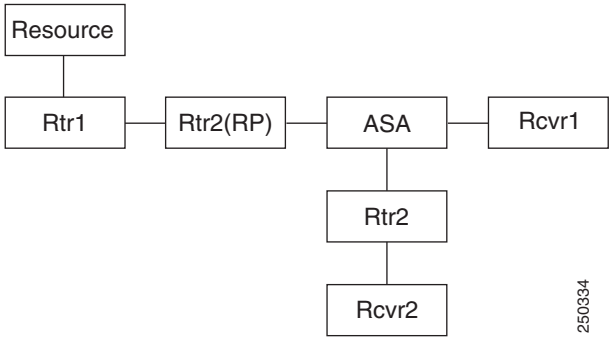


```
passwd password1
enable password password2
hostname myfirewall
asdm image disk0:/asdm.bin
boot system disk0:/image.bin
!
interface gigabitethernet 0/0
   nameif outside
   security-level 0
   ip address 10.1.1.2 255.255.255.0
   no shutdown
!
interface gigabitethernet 0/1
   description backup isp link
   nameif backupisp
   security-level 100
   ip address 172.16.2.2 255.255.255.0
   no shutdown
!
sla monitor 123
   type echo protocol ipIcmpEcho 10.2.1.2 interface outside
   num-packets 3
   timeout 1000
   frequency 3
sla monitor schedule 123 life forever start-time now
!
track 1 rtr 123 reachability
!
route outside 0.0.0.0 0.0.0.0 10.1.1.1 track 1
! The above route is used while the tracked object, router 10.2.1.2
! is available. It is removed when the router becomes unavailable.
!
route backupisp 0.0.0.0 0.0.0.0 172.16.2.1 254
! The above route is a floating static route that is added to the
! routing table when the tracked route is removed.
```

Example 8: Multicast Routing

This configuration shows a source that is sending out multicast traffic with two listeners that are watching for messages. A network lies between the source and the receivers, and all devices need to build up the PIM tree properly for the traffic to flow. This includes the ASA 5505 adaptive security appliance, and all IOS routers.

Figure A-8 Multicast Routing Configuration



Note Multicast routing only works in single routed mode.

- For PIM Sparse Mode, page A-22
- For PIM bidir Mode, page A-23

For PIM Sparse Mode

This configuration enables multicast routing for PIM Sparse Mode.

```
hostname asa
multicast-routing

interface GigabitEthernet0/0
 nameif outside
 security-level 0
 ip address 10.1.1.1 255.255.255.0

interface GigabitEthernet0/1
 nameif inside
 security-level 100
 ip address 10.1.2.1 255.255.255.0

interface GigabitEthernet0/2
 nameif dmz
 security-level 50
 ip address 10.1.3.1 255.255.255.0
 igmp join-group 227.1.2.3

! Specify the RP
pim rp-address 10.1.1.2

! Specify ACL configuration on the interfaces
access-list mcast permit pim any any
access-list mcast permit igmp any any
access-list mcast permit ospf any any
access-list mcast permit icmp any any
access-list mcast permit tcp any any eq 80
access-list mcast permit udp any 224.0.0.0 240.0.0.0

no failover
access-group mcast in interface outside
access-group mcast in interface inside
```

```
access-group mcast in interface dmz

!   Configures unicast routing
router ospf 1
 network 10.1.1.0 255.255.255.0 area 0
 network 10.1.2.0 255.255.255.0 area 0
 network 10.1.3.0 255.255.255.0 area 0
 log-adj-changes
!
```

For PIM bidir Mode

```
hostname asa
multicast-routing
!
interface GigabitEthernet0/0
 nameif outside
 security-level 0
 ip address 10.1.1.1 255.255.255.0
!
interface GigabitEthernet0/1
 nameif inside
 security-level 100
 ip address 10.1.2.1 255.255.255.0
!
interface GigabitEthernet0/2
 nameif dmz
 security-level 50
 ip address 10.1.3.1 255.255.255.0
 igmp join-group 227.1.2.3

! Specify the RP
pim rp-address 10.1.1.2 bidir


! Specify ACL configuration on the interfaces
access-list mcast permit pim any any
access-list mcast permit igmp any any
access-list mcast permit ospf any any
access-list mcast permit icmp any any
access-list mcast permit tcp any any eq 80
access-list mcast permit udp any 224.0.0.0 240.0.0.0

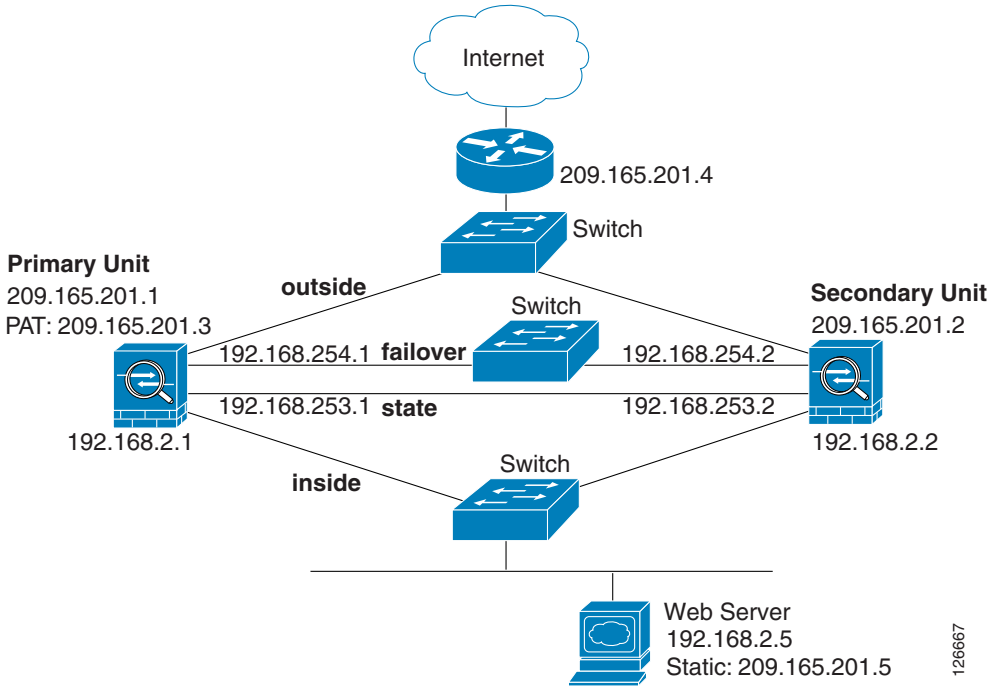
no failover
access-group mcast in interface outside
access-group mcast in interface inside
access-group mcast in interface dmz

!   Configures unicast routing
router ospf 1
 network 10.1.1.0 255.255.255.0 area 0
 network 10.1.2.0 255.255.255.0 area 0
 network 10.1.3.0 255.255.255.0 area 0
 log-adj-changes
```

Example 9: LAN-Based Active/Standby Failover (Routed Mode)

Figure A-9 shows the network diagram for a failover configuration using an Ethernet failover link. The units are configured to detect unit failures and to fail over in under a second (see the **failover polltime unit** command in the primary unit configuration).

Figure A-9 LAN-Based Failover Configuration



See the following sections for primary or secondary unit configuration scenarios:

- Primary Unit Configuration for Example 9, page A-24
- Secondary Unit Configuration for Example 9, page A-25

Primary Unit Configuration for Example 9

```
hostname pixfirewall
enable password myenablepassword
password mypassword
interface gigabitethernet0/0
   nameif outside
   ip address 209.165.201.1 255.255.255.224 standby 209.165.201.2
   no shutdown
interface gigabitethernet0/1
   nameif inside
   ip address 192.168.2.1 255.255.255.0 standby 192.168.2.2
   no shutdown
interface gigabitethernet0/2
   description LAN Failover Interface
   no shutdown
interface gigabitethernet0/3
   description STATE Failover Interface
```

```
telnet 192.168.2.45 255.255.255.255 inside
access-list acl_out permit tcp any host 209.165.201.5 eq 80
failover
failover lan unit primary
failover lan interface failover gigabitethernet0/2
failover lan enable
! The failover lan enable command is required on the PIX security appliance only.
failover polltime unit msec 200 holdtime msec 800
failover key key1
failover link state gigabitethernet0/3
failover interface ip failover 192.168.254.1 255.255.255.0 standby 192.168.254.2
failover interface ip state 192.168.253.1 255.255.255.0 standby 192.168.253.2
global (outside) 1 209.165.201.3 netmask 255.255.255.224
nat (inside) 1 0.0.0.0 0.0.0.0
static (inside,outside) 209.165.201.5 192.168.2.5 netmask 255.255.255.255 0 0
access-group acl_out in interface outside
route outside 0.0.0.0 0.0.0.0 209.165.201.4 1
```

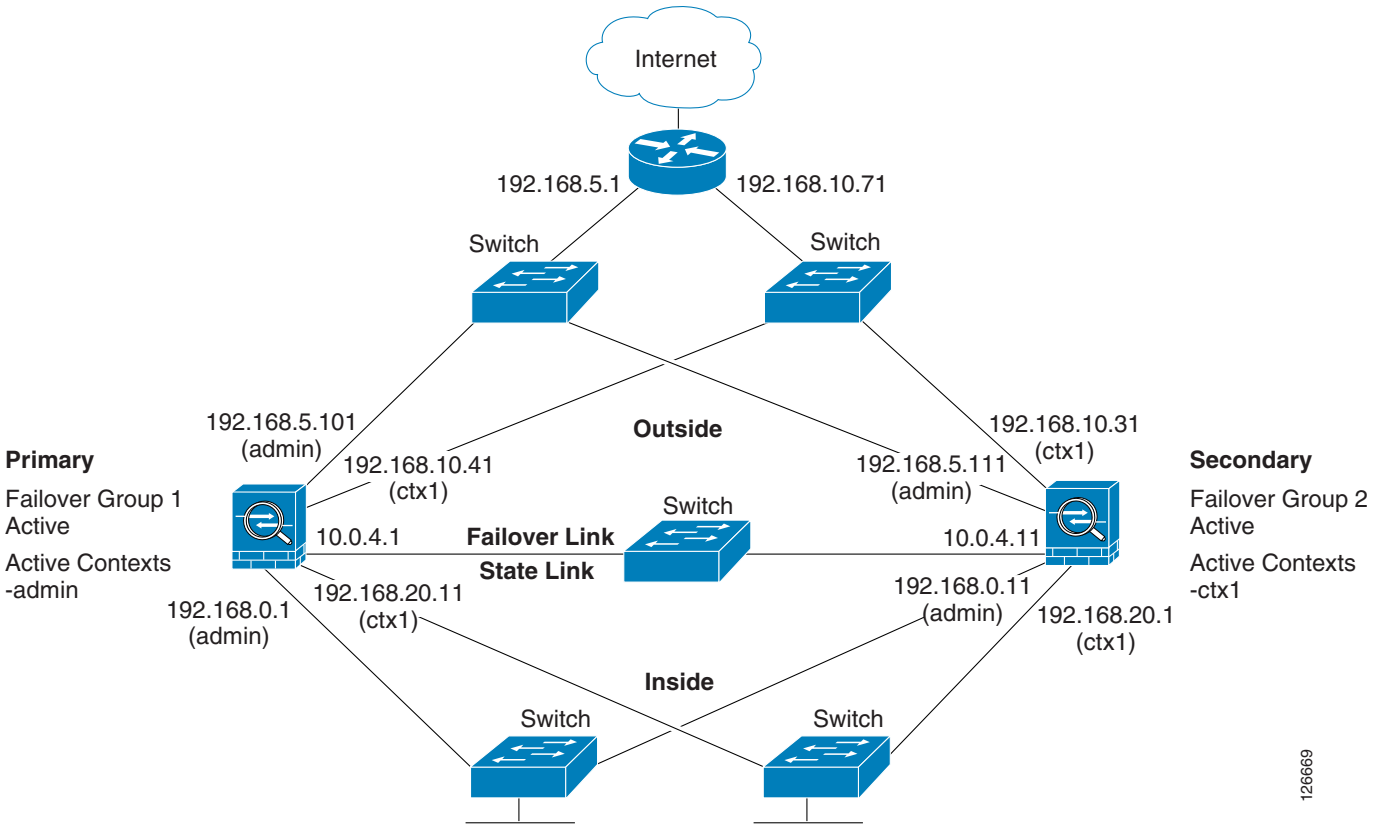
Secondary Unit Configuration for Example 9

```
failover
failover lan unit secondary
failover lan interface failover gigabitethernet0/2
failover lan enable
failover key key1
failover interface ip failover 192.168.254.1 255.255.255.0 standby 192.168.254.2
```

Example 10: LAN-Based Active/Active Failover (Routed Mode)

The following example shows how to configure Active/Active failover. In this example there are 2 user contexts, named admin and ctx1. Figure A-10 shows the network diagram for the example.

Figure A-10 Active/Active Failover Configuration



See the following sections for the configurations for this scenario:

- Primary Unit Configuration for Example 10, page A-26
- Secondary Unit Configuration for Example 10, page A-28

Primary Unit Configuration for Example 10

See the following sections for the primary unit configuration:

- Primary System Configuration for Example 10, page A-26
- Primary admin Context Configuration for Example 10, page A-27
- Primary ctx1 Context Configuration for Example 10, page A-28

Primary System Configuration for Example 10

You must first enable multiple context mode using the **mode multiple** command. The mode is not stored in the configuration file, even though it endures reboots. Enter the **show mode** command to view the current mode.

```
hostname ciscopix
enable password farscape
password crichton
asdm image flash:/asdm.bin
```

```
boot system flash:/cdisk.bin
mac-address auto
interface gigabitethernet0/0
   description LAN/STATE Failover Interface
interface gigabitethernet0/1
   no shutdown
interface gigabitethernet0/2
   no shutdown
interface gigabitethernet0/3
   no shutdown
interface gigabitethernet1/0
   no shutdown
interface gigabitethernet1/1
   no shutdown
interface gigabitethernet1/2
   no shutdown
interface gigabitethernet1/3
   no shutdown
failover
failover lan unit primary
failover lan interface folink gigabitethernet0/0
failover link folink gigabitethernet0/0
failover interface ip folink 10.0.4.1 255.255.255.0 standby 10.0.4.11
failover group 1
   primary
   preempt 60
failover group 2
   secondary
   preempt 60
admin-context admin
context admin
   description admin
   allocate-interface gigabitethernet0/1
   allocate-interface gigabitethernet0/2
   config-url flash:/admin.cfg
   join-failover-group 1
context ctx1
   description context 1
   allocate-interface gigabitethernet0/3
   allocate-interface gigabitethernet1/0
   config-url flash:/ctx1.cfg
   join-failover-group 2
```

Primary admin Context Configuration for Example 10

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
enable password frek
password elixir
hostname admin
interface gigabitethernet0/1
   nameif outside
   security-level 0
   ip address 192.168.5.101 255.255.255.0 standby 192.168.5.111
interface gigabitethernet0/2
   nameif inside
   security-level 100
   ip address 192.168.0.1 255.255.255.0 standby 192.168.0.11
monitor-interface outside
monitor-interface inside
route outside 0.0.0.0 0.0.0.0 192.168.5.1 1
```

```
ssh 192.168.0.2 255.255.255.255 inside
```

Primary ctx1 Context Configuration for Example 10

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
enable password quadrophenia
password tommy
hostname ctx1
interface gigabitethernet0/3
   nameif inside
   security-level 100
   ip address 192.168.20.1 255.255.255.0 standby 192.168.20.11
interface gigabitethernet1/0
   nameif outside
   security-level 0
   ip address 192.168.10.31 255.255.255.0 standby 192.168.10.41
   asr-group 1
access-list 201 extended permit ip any any
access-group 201 in interface outside
logging enable
logging console informational
monitor-interface inside
monitor-interface outside
route outside 0.0.0.0 0.0.0.0 192.168.10.71 1
```

Secondary Unit Configuration for Example 10

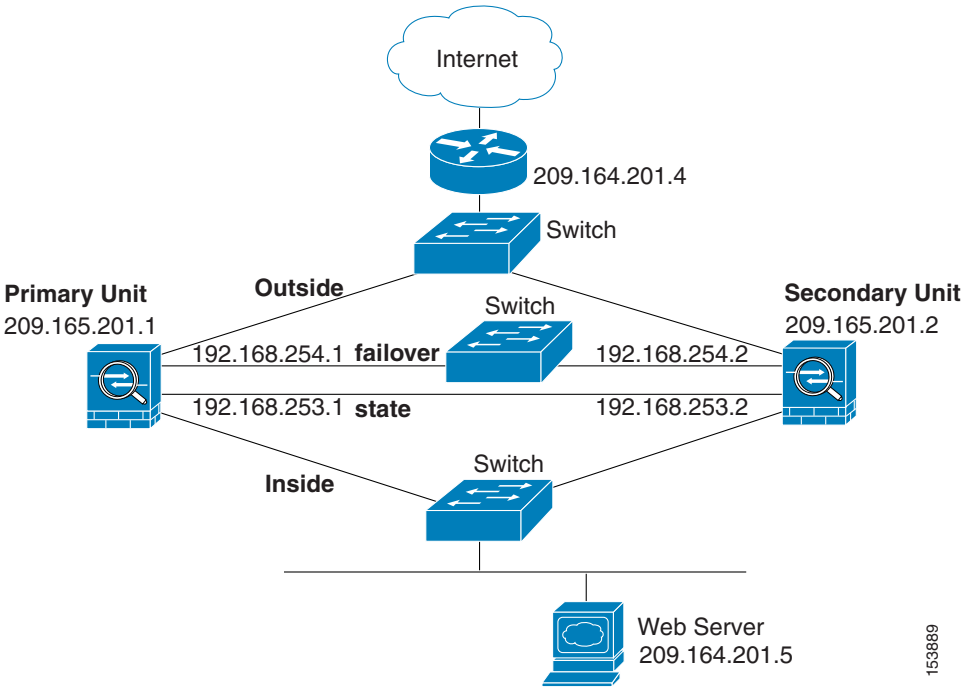
You only need to configure the secondary security appliance to recognize the failover link. The secondary security appliance obtains the context configurations from the primary security appliance upon booting or when failover is first enabled. The **preempt** commands in the failover group configurations cause the failover groups to become active on their designated unit after the configurations have been synchronized and the preempt delay has passed.

```
failover
failover lan unit secondary
failover lan interface folink gigabitethernet0/0
failover interface ip folink 10.0.4.1 255.255.255.0 standby 10.0.4.11
```

Example 11: LAN-Based Active/Standby Failover (Transparent Mode)

Figure A-11 shows the network diagram for a transparent mode failover configuration using an Ethernet failover link. The units are configured to detect unit failures and to fail over in under a second (see the **failover polltime unit** command in the primary unit configuration).

Figure A-11 Transparent Mode LAN-Based Failover Configuration



See the following sections for the configurations for this scenario:

- Primary Unit Configuration for Example 11, page A-29
- Secondary Unit Configuration for Example 11, page A-30

Primary Unit Configuration for Example 11

```
firewall transparent
hostname pixfirewall
enable password myenablepassword
password mypassword
interface gigabitethernet0/0
   nameif outside
   no shutdown
interface gigabitethernet0/1
   nameif inside
   no shutdown
interface gigabitethernet0/2
   description LAN Failover Interface
   no shutdown
interface gigabitethernet0/3
   description STATE Failover Interface
telnet 192.168.2.45 255.255.255.255 inside
access-list acl_out permit tcp any host 209.165.201.5 eq 80
ip address 209.165.201.1 255.255.255.0 standby 209.165.201.2
failover
failover lan unit primary
failover lan interface failover gigabitethernet0/2
failover lan enable
! The failover lan enable command is required on the PIX security appliance only.
failover polltime unit msec 200 holdtime msec 800
```

```
failover key key1
failover link state gigabitethernet0/3
failover interface ip failover 192.168.254.1 255.255.255.0 standby 192.168.254.2
failover interface ip state 192.168.253.1 255.255.255.0 standby 192.168.253.2
access-group acl_out in interface outside
route outside 0.0.0.0 0.0.0.0 209.165.201.4 1
```

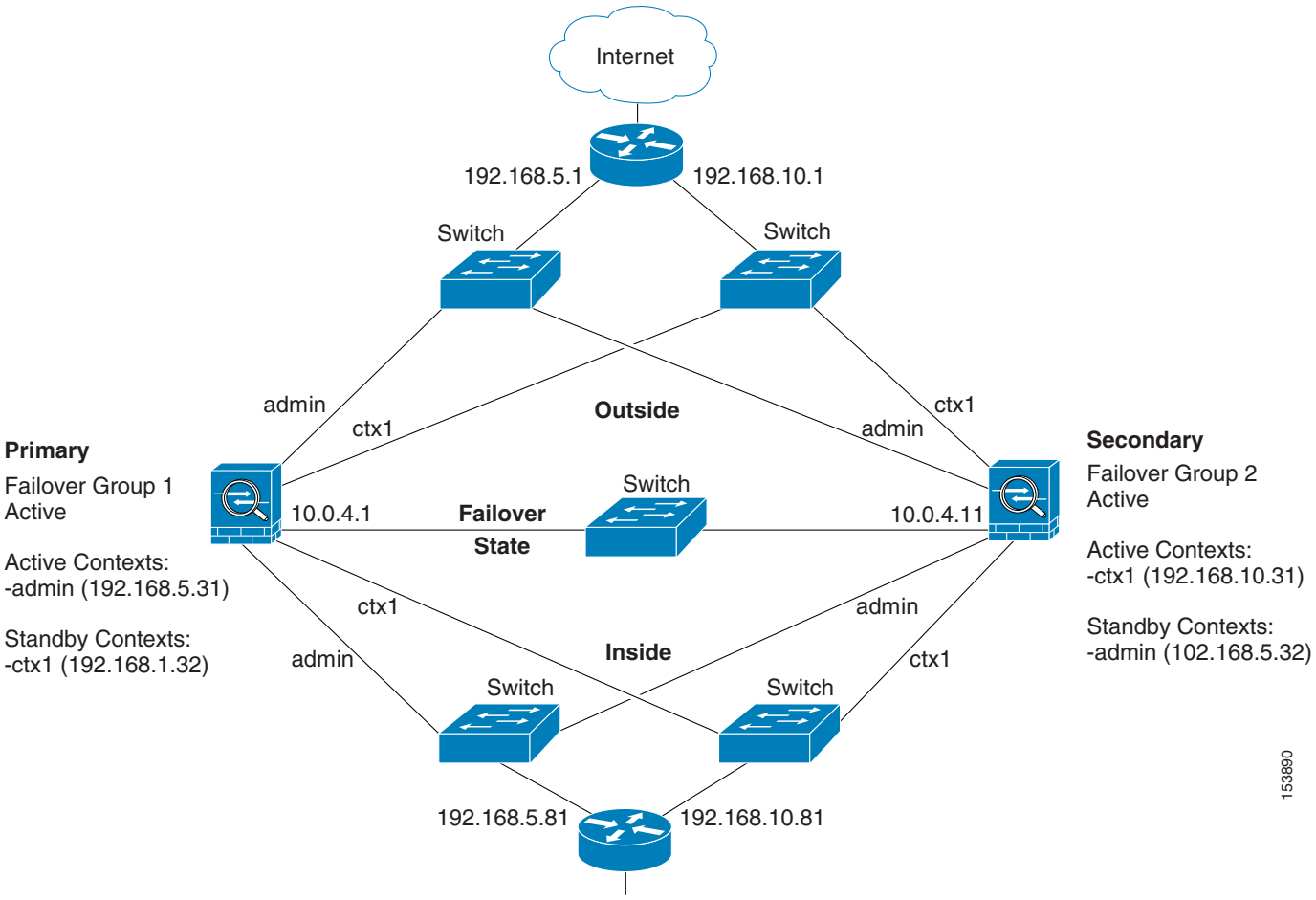
Secondary Unit Configuration for Example 11

```
firewall transparent
failover
failover lan unit secondary
failover lan interface failover gigabitethernet0/2
failover lan enable
failover key key1
failover interface ip failover 192.168.254.1 255.255.255.0 standby 192.168.254.2
```

Example 12: LAN-Based Active/Active Failover (Transparent Mode)

The following example shows how to configure transparent mode Active/Active failover. In this example there are 2 user contexts, named admin and ctx1. Figure A-12 shows the network diagram for the example.

Figure A-12 Transparent Mode Active/Active Failover Configuration



See the following sections for the configurations for this scenario:

- Primary Unit Configuration for Example 12, page A-31
- Secondary Unit Configuration for Example 12, page A-33

Primary Unit Configuration for Example 12

See the following sections for the primary unit configuration:

- Primary System Configuration for Example 12, page A-31
- Primary admin Context Configuration for Example 12, page A-32
- Primary ctx1 Context Configuration for Example 12, page A-33

Primary System Configuration for Example 12

You must first enable multiple context mode using the **mode multiple** command. The mode is not stored in the configuration file, even though it endures reboots. Enter the **show mode** command to view the current mode.

```
firewall transparent
```

```
hostname ciscopix
enable password farscape
password crichton
asdm image flash:/asdm.bin
boot system flash:/cdisk.bin
mac-address auto
interface gigabitethernet0/0
   description LAN/STATE Failover Interface
interface gigabitethernet0/1
   no shutdown
interface gigabitethernet0/2
   no shutdown
interface gigabitethernet0/3
   no shutdown
interface gigabitethernet1/0
   no shutdown
interface gigabitethernet1/1
   no shutdown
interface gigabitethernet1/2
   no shutdown
interface gigabitethernet1/3
   no shutdown
failover
failover lan unit primary
failover lan interface folink gigabitethernet0/0
failover link folink gigabitethernet0/0
failover interface ip folink 10.0.4.1 255.255.255.0 standby 10.0.4.11
failover group 1
   primary
   preempt
failover group 2
   secondary
   preempt
admin-context admin
context admin
   description admin
   allocate-interface gigabitethernet0/1
   allocate-interface gigabitethernet0/2
   config-url flash:/admin.cfg
   join-failover-group 1
context ctx1
   description context 1
   allocate-interface gigabitethernet0/3
   allocate-interface gigabitethernet1/0
   config-url flash:/ctx1.cfg
   join-failover-group 2
```

Primary admin Context Configuration for Example 12

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
enable password frek
password elixir
hostname admin
interface gigabitethernet0/1
   nameif outside
   security-level 0
interface gigabitethernet0/2
   nameif inside
   security-level 100
ip address 192.168.5.31 255.255.255.0 standby 192.168.5.32
```

```
monitor-interface outside
monitor-interface inside
route outside 0.0.0.0 0.0.0.0 192.168.5.1 1
ssh 192.168.5.72 255.255.255.255 inside
```

Primary ctx1 Context Configuration for Example 12

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
enable password quadrophenia
password tommy
hostname ctx1
interface gigabitethernet0/3
   nameif inside
   security-level 100
interface gigabitethernet1/0
   nameif outside
   security-level 0
access-list 201 extended permit ip any any
access-group 201 in interface outside
logging enable
logging console informational
ip address 192.168.10.31 255.255.255.0 standby 192.168.10.32
monitor-interface inside
monitor-interface outside
route outside 0.0.0.0 0.0.0.0 192.168.10.1 1
```

Secondary Unit Configuration for Example 12

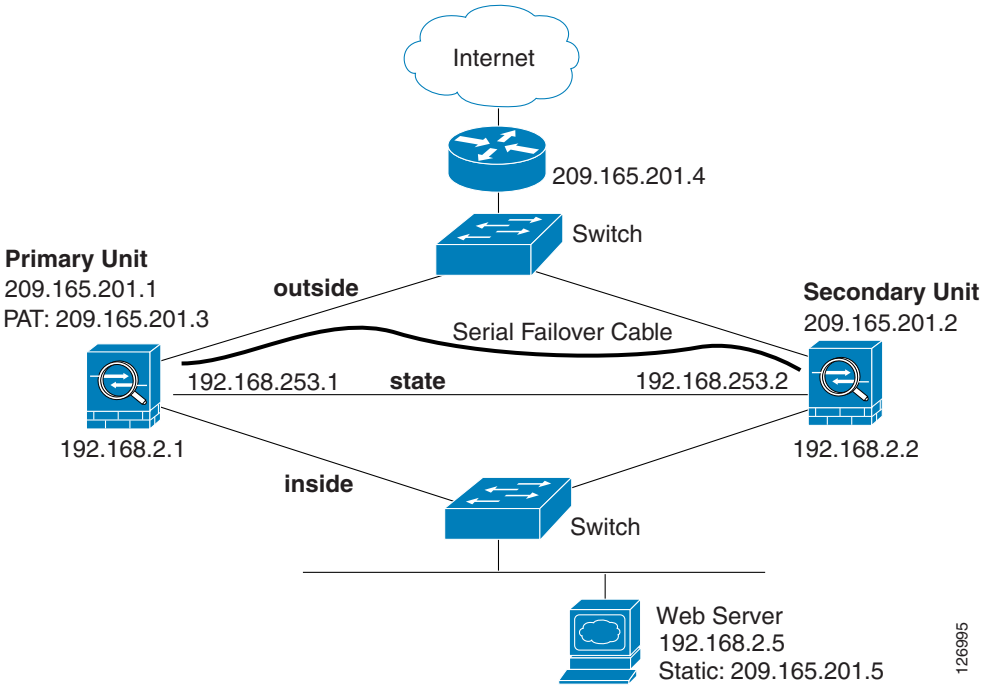
You only need to configure the secondary security appliance to recognize the failover link. The secondary security appliance obtains the context configurations from the primary security appliance upon booting or when failover is first enabled. The **preempt** commands in the failover group configurations cause the failover groups to become active on their designated unit after the configurations have been synchronized and the preempt delay has passed.

```
firewall transparent
failover
failover lan unit secondary
failover lan interface folink gigabitethernet0/0
failover interface ip folink 10.0.4.1 255.255.255.0 standby 10.0.4.11
```

Example 13: Cable-Based Active/Standby Failover (Routed Mode)

Figure A-13 shows the network diagram for a failover configuration using a serial Failover cable. This configuration is only available on the PIX security appliance. This example also specifies a stateful failover configuration.

Figure A-13 Cable-Based Failover Configuration



The following are the typical commands in a cable-based failover configuration.

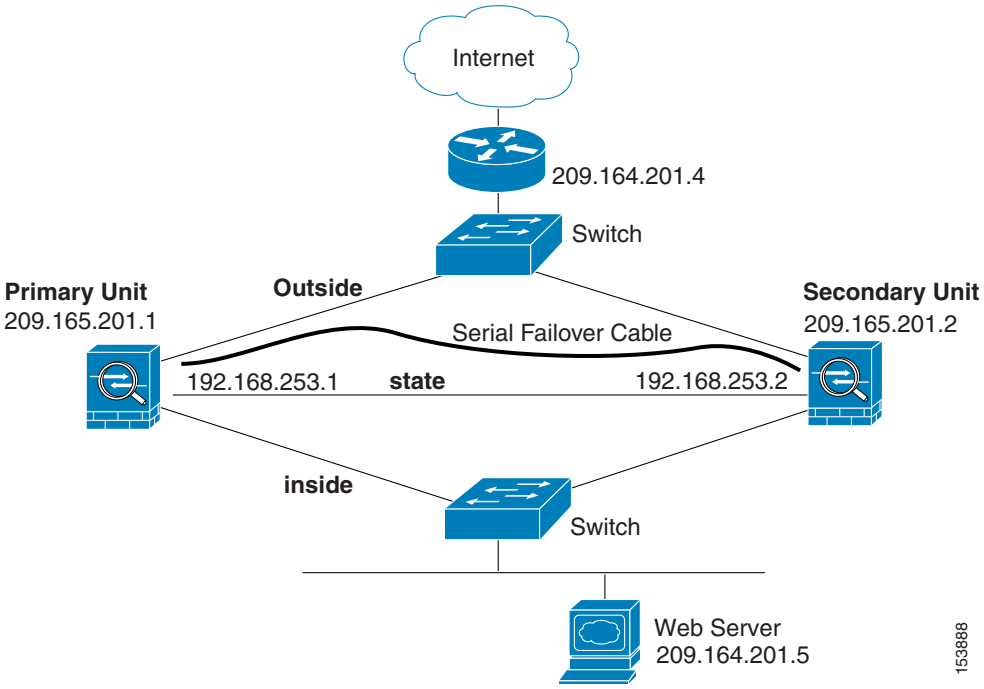
```
enable password myenablepassword
passwd mypassword
hostname pixfirewall
asdm image flash:/asdm.bin
boot system flash:/image.bin
interface Ethernet0
   nameif outside
   security-level 0
   speed 100
   duplex full
   ip address 209.165.201.1 255.255.255.224 standby 209.165.201.2
   no shutdown
interface Ethernet1
   nameif inside
   security-level 100
   speed 100
   duplex full
   ip address 192.168.2.1 255.255.255.0 standby 192.168.2.2
   no shutdown
interface Ethernet3
   description STATE Failover Interface
```

```
telnet 192.168.2.45 255.255.255.255 inside
access-list acl_in permit tcp any host 209.165.201.5 eq 80
access-group acl_in in interface outside
failover
! Enables cable-based failover on the PIX security appliance
failover link state Ethernet3
failover interface ip state 192.168.253.1 255.255.255.252 standby 192.168.253.2
! The previous two lines are necessary for a stateful failover
global (outside) 1 209.165.201.3 netmask 255.255.255.224
nat (inside) 1 0.0.0.0 0.0.0.0
static (inside,outside) 209.165.201.5 192.168.2.5 netmask 255.255.255.255 0 0
route outside 0.0.0.0 0.0.0.0 209.165.201.4 1
```

Example 14: Cable-Based Active/Standby Failover (Transparent Mode)

Figure A-14 shows the network diagram for a transparent mode failover configuration using a serial Failover cable. This configuration is only available on the PIX 500 series security appliance.

Figure A-14 Transparent Mode Cable-Based Failover Configuration



The following are the typical commands in a cable-based, transparent firewall failover configuration.

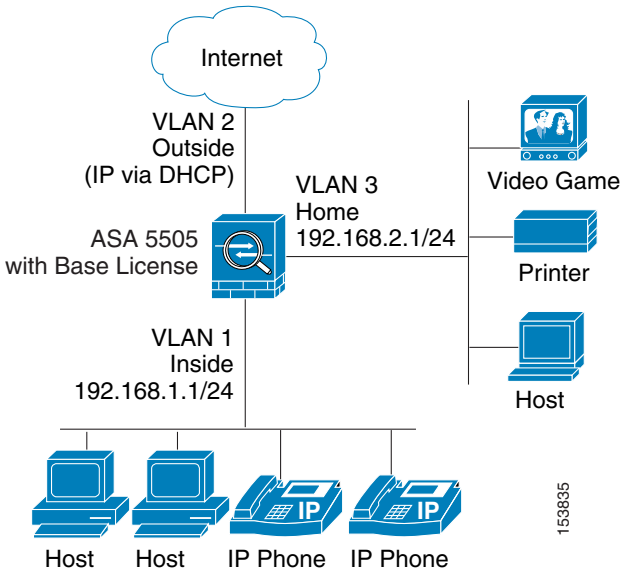
```
enable password myenablepassword
passwd mypassword
hostname pixfirewall
asdm image flash:/asdm.bin
boot system flash:/image.bin
firewall transparent
interface Ethernet0
```

```
   speed 100
   duplex full
   nameif outside
   security-level 0
   no shutdown
interface Ethernet1
   speed 100
   duplex full
   nameif inside
   security-level 100
   no shutdown
interface Ethernet3
   description STATE Failover Interface
telnet 192.168.2.45 255.255.255.255 mgmt
access-list acl_in permit tcp any host 209.165.201.5 eq 80
access-group acl_in in interface outside
ip address 209.165.201.1 255.255.255.0 standby 209.165.201.2
failover
failover link state Ethernet3
failover interface ip state 192.168.253.1 255.255.255.0 standby 192.168.253.2
route outside 0.0.0.0 0.0.0.0 209.165.201.4 1
```

Example 15: ASA 5505 Base License

This configuration creates three VLANs: inside (business), outside (Internet), and home (see Figure A-15). Both the home and inside VLANs can access the outside, but the home VLAN cannot access the inside VLAN. The inside VLAN can access the home VLAN so both VLANs can share a printer. Because the outside IP address is set using DHCP, the inside and home VLANs use interface PAT when accessing the Internet.

Figure A-15 ASA 5505 Base License



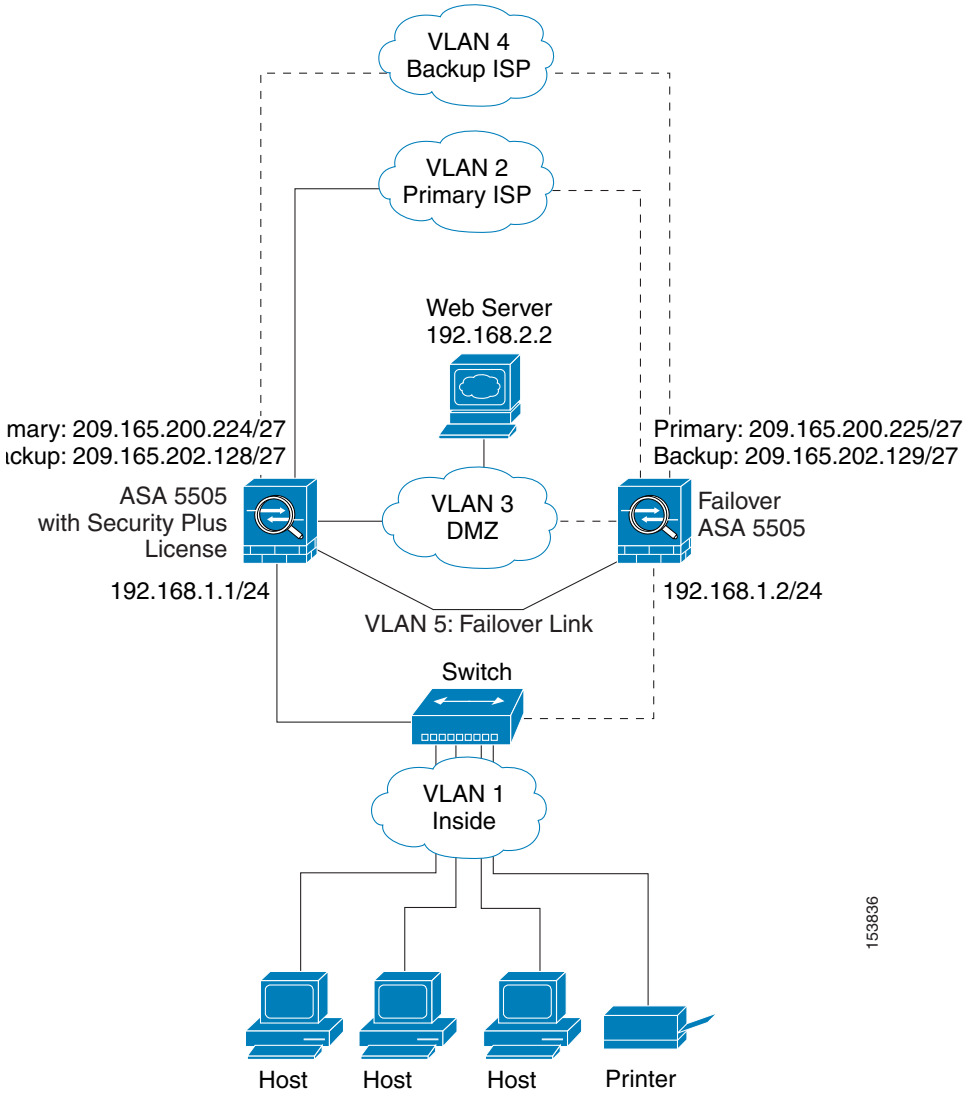

```
passwd g00fba11
enable password gen1u$
hostname Buster
```

```
asdm image disk0:/asdm.bin
boot system disk0:/image.bin
interface vlan 2
   nameif outside
   security-level 0
   ip address dhcp setroute
   no shutdown
interface vlan 1
   nameif inside
   security-level 100
   ip address 192.168.1.1 255.255.255.0
   no shutdown
interface vlan 3
! This interface cannot communicate with the inside interface. This is required using
! the Base license
   no forward interface vlan 1
   nameif home
   security-level 50
   ip address 192.168.2.1 255.255.255.0
   no shutdown
interface ethernet 0/0
   switchport access vlan 2
   no shutdown
   interface ethernet 0/1
   switchport access vlan 1
   no shutdown
interface ethernet 0/2
   switchport access vlan 1
   no shutdown
interface ethernet 0/3
   switchport access vlan 3
   no shutdown
interface ethernet 0/4
   switchport access vlan 3
   no shutdown
interface ethernet 0/5
   switchport access vlan 3
   no shutdown
interface ethernet 0/6
   description PoE for IP phone1
   switchport access vlan 1
   no shutdown
interface ethernet 0/7
   description PoE for IP phone2
   switchport access vlan 1
   no shutdown
nat (inside) 1 0 0
nat (home) 1 0 0
global (outside) 1 interface
! The previous NAT statements match all addresses on inside and home, so you need to
! also perform NAT when hosts access the inside or home networks (as well as the outside).
! Or you can exempt hosts from NAT for inside <--> home traffic, as effected by the
! following:
access-list natexmpt-inside extended permit ip any 192.168.2.0 255.255.255.0
access-list natexmpt-home extended permit ip any 192.168.1.0 255.255.255.0
nat (inside) 0 access-list natexmpt-inside
nat (home) 0 access-list natexmpt-home
http server enable
http 192.168.1.0 255.255.255.0 inside
dhcpd address 192.168.1.2-192.168.1.254 inside
dhcpd auto_config outside
dhcpd enable inside
logging asdm informational
ssh 192.168.1.0 255.255.255.0 inside
```

Example 16: ASA 5505 Security Plus License with Failover and Dual-ISP Backup

This configuration creates five VLANs: inside, outside, dmz, backup-isp and faillink (see Figure A-16).

Figure A-16 ASA 5505 Security Plus License with Failover and Dual-ISP Backup



See the following sections for the configurations for this scenario:

- Primary Unit Configuration for Example 16, page A-38
- Secondary Unit Configuration for Example 16, page A-40

Primary Unit Configuration for Example 16

```
passwd g00fba11
enable password gen1u$
```

```
hostname Buster
asdm image disk0:/asdm.bin
boot system disk0:/image.bin
interface vlan 2
   description Primary ISP interface
   nameif outside
   security-level 0
   ip address 209.165.200.224 standby 209.165.200.225
   backup interface vlan 4
   no shutdown
interface vlan 1
   nameif inside
   security-level 100
   ip address 192.168.1.1 255.255.255.0
   no shutdown
interface vlan 3
   nameif dmz
   security-level 50
   ip address 192.168.2.1 255.255.255.0
   no shutdown
interface vlan 4
   description Backup ISP interface
   nameif backup-isp
   security-level 0
   ip address 209.168.202.128 standby 209.168.202.129
   no shutdown
interface vlan 5
   description LAN Failover Interface
interface ethernet 0/0
   switchport access vlan 2
   no shutdown
interface ethernet 0/1
   switchport access vlan 4
   no shutdown
interface ethernet 0/2
   switchport access vlan 1
   no shutdown
interface ethernet 0/3
   switchport access vlan 3
   no shutdown
interface ethernet 0/4
   switchport access vlan 5
   no shutdown
failover
failover lan unit primary
failover lan interface faillink vlan5
failover lan faillink vlan5
failover polltime unit 3 holdtime 10
failover key key1
failover interface ip faillink 10.1.1.1 255.255.255.0 standby 10.1.1.2
nat (inside) 1 0 0
nat (home) 1 0 0
global (outside) 1 interface
! The previous NAT statements match all addresses on inside and home, so you need to
! also perform NAT when hosts access the inside or home networks (as well as the outside).
! Or you can exempt hosts from NAT for inside <--> home traffic, as effected by the
! following:
access-list natexmpt-inside extended permit ip any 192.168.2.0 255.255.255.0
access-list natexmpt-home extended permit ip any 192.168.1.0 255.255.255.0
nat (inside) 0 access-list natexmpt-inside
nat (home) 0 access-list natexmpt-home
sla monitor 123
 type echo protocol ipIcmpEcho 209.165.200.234 interface outside
 num-packets 2
```

```
 frequency 5
sla monitor schedule 123 life forever start-time now
track 1 rtr 123 reachability
route outside 0 0 209.165.200.234 1 track 1
! This route is for the primary ISP.
route backup-isp 0 0 209.165.202.154 2
! If the link goes down for the primary ISP, either due to a hardware failure
! or unplugged cable, then this route will be used.
http server enable
http 192.168.1.0 255.255.255.0 inside
dhcpd address 192.168.1.2-192.168.1.254 inside
dhcpd auto_config outside
dhcpd enable inside
logging asdm informational
ssh 192.168.1.0 255.255.255.0 inside
```

Secondary Unit Configuration for Example 16

You only need to configure the secondary security appliance to recognize the failover link. The secondary security appliance obtains the context configurations from the primary security appliance upon booting or when failover is first enabled.

```
interface ethernet 0/4
   switchport access vlan 5
   no shutdown
failover
failover lan unit secondary
failover lan interface faillink vlan5
failover polltime unit 3 holdtime 10
failover key key1
failover interface ip faillink 10.1.1.1 255.255.255.0 standby 10.1.1.2
```

Example 17: AIP SSM in Multiple Context Mode

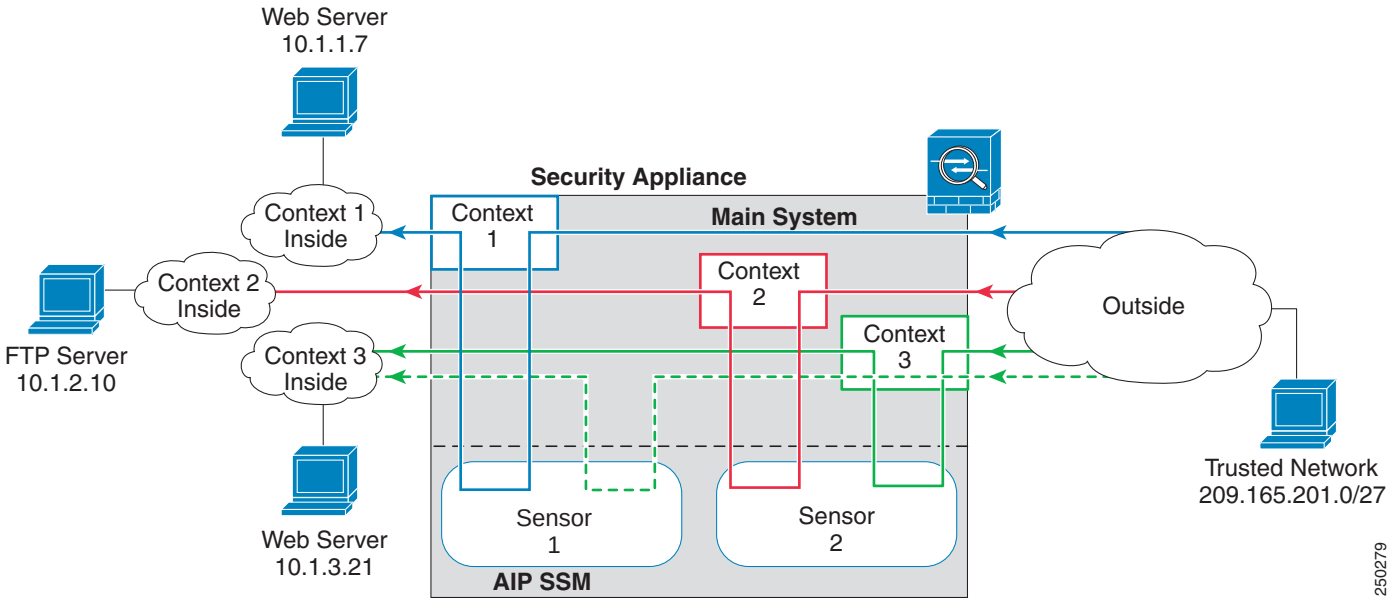
This configuration assigns two virtual IPS sensors to three contexts. Context 1 uses sensor 1, context 2 uses sensor 2 (for greater security), and context 3 uses sensor 2 for most traffic, but uses sensor 1 for a more trusted outside network (see Figure A-17).

For Context 1, only the trusted network is allowed to access a web server and manage the context using SSH.

For Context 2, any outside user can access the FTP server.

For Context 3, any outside user can access the web server, but the trusted network can access anything on the inside network.

Figure A-17 Security Contexts and Virtual Sensors



See the following sections for the configurations for this scenario:

- System Configuration for Example 17, page A-41
- Context 1 Configuration for Example 17, page A-42
- Context 2 Configuration for Example 17, page A-42
- Context 3 Configuration for Example 17, page A-43

System Configuration for Example 17

You must first enable multiple context mode using the **mode multiple** command. The mode is not stored in the configuration file, even though it endures reboots. Enter the **show mode** command to view the current mode.

```
hostname Farscape
password passw0rd
enable password chr1cht0n
mac-address auto
asdm image disk0:/asdm.bin
boot system disk0:/image.bin
admin-context context 1
interface gigabitethernet 0/0
   no shutdown
interface gigabitethernet 0/1
   no shutdown
interface gigabitethernet 0/2
   no shutdown
interface gigabitethernet 0/3
   no shutdown
context 1
   allocate-interface gigabitethernet0/0
   allocate-interface gigabitethernet0/3
   allocate-ips sensor1
   config-url ftp://user1:passw0rd@10.1.1.1/configlets/context1.cfg
context 2
```

```
   allocate-interface gigabitethernet0/1
   allocate-interface gigabitethernet0/3
   allocate-ips sensor2
   config-url ftp://user1:passw0rd@10.1.1.1/configlets/context2.cfg
context 3
   allocate-interface gigabitethernet0/2
   allocate-interface gigabitethernet0/3
   allocate-ips sensor1
   allocate-ips sensor2 default
   config-url ftp://user1:passw0rd@10.1.1.1/configlets/context3.cfg
```

Context 1 Configuration for Example 17

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
hostname context1
domain example.com
interface gigabitethernet 0/3
   nameif outside
   security-level 0
   ip address 209.165.200.225 255.255.255.224
   no shutdown
interface gigabitethernet 0/0
   nameif inside
   security-level 100
   ip address 10.1.1.1 255.255.255.0
   no shutdown
passwd seaandsand
enable password pinballwizard
route outside 0 0 209.165.200.230 1
ssh 209.165.201.0 255.255.255.224 outside
nat (inside) 1 10.1.1.0 255.255.255.0
global (outside) 1 209.165.200.252
! Trusted network can access the web server at 10.1.1.7
access-list INBOUND extended permit tcp 209.165.201.0 255.255.255.224 host 10.1.1.7 eq
http
access-group INBOUND in interface outside
! Any traffic allowed to the inside of context 1 must go through
! the IPS sensor assigned to the context.
! Traffic is in promiscuous mode (a separate stream is sent
! to the IPS sensor. If the sensor fails, traffic continues.
access-list IPS extended permit ip any any
class-map my-ips-class
   match access-list IPS
policy-map my-ips-policy
   class my-ips-class
      ips promiscous fail-open
service-policy my-ips-policy interface outside
```

Context 2 Configuration for Example 17

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
hostname context2
domain example.com
interface gigabitethernet 0/3
   nameif outside
```

```
   security-level 0
   ip address 209.165.200.226 255.255.255.224
   no shutdown
interface gigabitethernet 0/1
   nameif inside
   security-level 100
   ip address 10.1.2.1 255.255.255.0
   no shutdown
passwd drjimmy
enable password acidqueen
route outside 0 0 209.165.200.230 1
ssh 10.1.2.67 255.255.255.255 inside
nat (inside) 1 10.1.2.0 255.255.255.0
global (outside) 1 209.165.200.253
! All users can access the FTP server at 10.1.2.10
access-list FTP extended permit tcp any any eq ftp
access-group FTP in interface outside
! Any traffic allowed to the inside of context 2 must go through
! the IPS sensor assigned to the context.
! Traffic is in inline mode (traffic is sent
! to the IPS sensor before continuing to the inside.)
! If the sensor fails, traffic stops.
access-list IPS permit ip any any
class-map my-ips-class
   match access-list IPS
policy-map my-ips-policy
   class my-ips-class
      ips inline fail-close
service-policy my-ips-policy interface outside
```

Context 3 Configuration for Example 17

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
hostname context3
domain example.com
interface gigabitethernet 0/3
   nameif outside
   security-level 0
   ip address 209.165.200.227 255.255.255.224
   no shutdown
interface gigabitethernet 0/1
   nameif inside
   security-level 100
   ip address 10.1.3.1 255.255.255.0
   no shutdown
passwd lovereign
enable password underture
route outside 0 0 209.165.200.230 1
ssh 209.165.201.0 255.255.255.224 outside
nat (inside) 1 10.1.3.0 255.255.255.0
global (outside) 1 209.165.200.254
! All users can access the web server at 10.1.3.21
! The trusted network 209.165.201.0/27 can access all of the inside nw.
access-list IN_CONTEXT3 extended permit ip 209.165.201.0 255.255.255.224 any
access-list IN_CONTEXT3 extended permit tcp any host 10.1.3.21 eq http
access-group IN_CONTEXT3 in interface outside
! Traffic from 209.165.201.0/27 goes through IPS sensor 1.
! Traffic is in promiscuous mode (a separate stream is sent
! to the IPS sensor. If the sensor fails, traffic continues.
```

```
! All other traffic allowed to the inside of context 1 must go
! through sensor 2. Traffic is in inline mode (traffic is sent
! to the IPS sensor before continuing to the inside.)
! If the sensor fails, traffic stops.
access-list my-ips-acl permit ip 209.165.201.0 255.255.255.224 any
class-map my-ips-class
   match access-list my-ips-acl
access-list my-ips-acl2 permit ip any any
class-map my-ips-class2
   match access-list my-ips-acl2
policy-map my-ips-policy
   class my-ips-class
      ips promiscuous fail-open sensor sensor1
   class my-ips-class2
      ips inline fail-close sensor sensor2
service-policy my-ips-policy interface outside
```



APPENDIX B

Using the Command-Line Interface

This appendix describes how to use the CLI on the security appliance, and includes the following sections:

- Firewall Mode and Security Context Mode, page B-1
- Command Modes and Prompts, page B-2
- Syntax Formatting, page B-3
- Abbreviating Commands, page B-3
- Command-Line Editing, page B-3
- Command Completion, page B-4
- Command Help, page B-4
- Filtering show Command Output, page B-4
- Command Output Paging, page B-6
- Adding Comments, page B-7
- Text Configuration Files, page B-7
- Supported Character Sets, page B-9

Note The CLI uses similar syntax and other conventions to the Cisco IOS CLI, but the security appliance operating system is not a version of Cisco IOS software. Do not assume that a Cisco IOS CLI command works with or has the same function on the security appliance.

Firewall Mode and Security Context Mode

The security appliance runs in a combination of the following modes:

- Transparent firewall or routed firewall mode

  The firewall mode determines if the security appliance runs as a Layer 2 or Layer 3 firewall.

- Multiple context or single context mode

  The security context mode determines if the security appliance runs as a single device or as multiple security contexts, which act like virtual devices.

Some commands are only available in certain modes.

Command Modes and Prompts

The security appliance CLI includes command modes. Some commands can only be entered in certain modes. For example, to enter commands that show sensitive information, you need to enter a password and enter a more privileged mode. Then, to ensure that configuration changes are not entered accidentally, you have to enter a configuration mode. All lower commands can be entered in higher modes, for example, you can enter a privileged EXEC command in global configuration mode.



Note The various types of prompts are all default prompts and when configured, they can be different.

- When you are in the system configuration or in single context mode, the prompt begins with the hostname:

  ```
  hostname
  ```

- When printing the prompt string, the prompt configuration is parsed and the configured keyword values are printed in the order in which you have set the **prompt** command. The keyword arguments can be any of the following and in any order: hostname, domain, context, priority, state.

  ```
  asa(config)# prompt hostname context priority state
  ```

- When you are within a context, the prompt begins with the hostname followed by the context name:

  ```
  hostname/context
  ```

The prompt changes depending on the access mode:

- User EXEC mode

  User EXEC mode lets you see minimum security appliance settings. The user EXEC mode prompt appears as follows when you first access the security appliance:

  ```
  hostname>

  hostname/context>
  ```

- Privileged EXEC mode

  Privileged EXEC mode lets you see all current settings up to your privilege level. Any user EXEC mode command will work in privileged EXEC mode. Enter the **enable** command in user EXEC mode, which requires a password, to start privileged EXEC mode. The prompt includes the number sign (#):

  ```
  hostname#

  hostname/context#
  ```

- Global configuration mode

  Global configuration mode lets you change the security appliance configuration. All user EXEC, privileged EXEC, and global configuration commands are available in this mode. Enter the **configure terminal** command in privileged EXEC mode to start global configuration mode. The prompt changes to the following:

  ```
  hostname(config)#

  hostname/context(config)#
  ```

- Command-specific configuration modes

From global configuration mode, some commands enter a command-specific configuration mode. All user EXEC, privileged EXEC, global configuration, and command-specific configuration commands are available in this mode. For example, the **interface** command enters interface configuration mode. The prompt changes to the following:

```
hostname(config-if)#

hostname/context(config-if)#
```

Syntax Formatting

Command syntax descriptions use the following conventions:

Table B-1 Syntax Conventions

Convention	Description
bold	Bold text indicates commands and keywords that you enter literally as shown.
italics	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional element (keyword or argument).
|	A vertical bar indicates a choice within an optional or required set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested sets of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.

Abbreviating Commands

You can abbreviate most commands down to the fewest unique characters for a command; for example, you can enter **wr t** to view the configuration instead of entering the full command **write terminal**, or you can enter **en** to start privileged mode and **conf t** to start configuration mode. In addition, you can enter **0** to represent **0.0.0.0**.

Command-Line Editing

The security appliance uses the same command-line editing conventions as Cisco IOS software. You can view all previously entered commands with the **show history** command or individually with the up arrow or **^p** command. Once you have examined a previously entered command, you can move forward in the list with the down arrow or **^n** command. When you reach a command you wish to reuse, you can edit it or press the **Enter** key to start it. You can also delete the word to the left of the cursor with **^w**, or erase the line with **^u**.

The security appliance permits up to 512 characters in a command; additional characters are ignored.

Command Completion

To complete a command or keyword after entering a partial string, press the **Tab** key. The security appliance only completes the command or keyword if the partial string matches only one command or keyword. For example, if you enter **s** and press the **Tab** key, the security appliance does not complete the command because it matches more than one command. However, if you enter **dis**, the **Tab** key completes the command **disable**.

Command Help

Help information is available from the command line by entering the following commands:

- **help** *command_name*

  Shows help for the specific command.

- *command_name* **?**

  Shows a list of arguments available.

- *string***?** (no space)

  Lists the possible commands that start with the string.

- **?** and **+?**

  Lists all commands available. If you enter **?**, the security appliance shows only commands available for the current mode. To show all commands available, including those for lower modes, enter **+?**.

Note If you want to include a question mark (?) in a command string, you must press **Ctrl-V** before typing the question mark so you do not inadvertently invoke CLI help.

Filtering show Command Output

You can use the vertical bar (|) with any **show** command and include a filter option and filtering expression. The filtering is performed by matching each output line with a regular expression, similar to Cisco IOS software. By selecting different filter options you can include or exclude all output that matches the expression. You can also display all output beginning with the line that matches the expression.

The syntax for using filtering options with the **show** command is as follows:

```
hostname# show command | {include | exclude | begin | grep [-v]} regexp
```

In this command string, the first vertical bar (|) is the operator and must be included in the command. This operator directs the output of the **show** command to the filter. In the syntax diagram, the other vertical bars (|) indicate alternative options and are not part of the command.

The **include** option includes all output lines that match the regular expression. The **grep** option without **-v** has the same effect. The **exclude** option excludes all output lines that match the regular expression. The **grep** option with **-v** has the same effect. The **begin** option shows all the output lines starting with the line that matches the regular expression.

Replace *regexp* with any Cisco IOS regular expression. The regular expression is not enclosed in quotes or double-quotes, so be careful with trailing white spaces, which will be taken as part of the regular expression.

When creating regular expressions, you can use any letter or number that you want to match. In addition, certain keyboard characters called *metacharacters* have special meaning when used in regular expressions.

Use **Ctrl+V** to escape all of the special characters in the CLI, such as question mark (?) or a tab. For example, type **d[Ctrl+V]?g** to enter **d?g** in the configuration.

Table B-2 lists the metacharacters that have special meanings.

Table B-2 regex Metacharacters

Character	Description	Notes
.	Dot	Matches any single character. For example, **d.g** matches dog, dag, dtg, and any word that contains those characters, such as doggonnit.
(*exp*)	Subexpression	A subexpression segregates characters from surrounding characters, so that you can use other metacharacters on the subexpression. For example, **d(o|a)g** matches dog and dag, but **do|ag** matches do and ag. A subexpression can also be used with repeat quantifiers to differentiate the characters meant for repetition. For example, **ab(xy){3}z** matches abxyxyxyz.
|	Alternation	Matches either expression it separates. For example, **dog|cat** matches dog or cat.
?	Question mark	A quantifier that indicates that there are 0 or 1 of the previous expression. For example, **lo?se** matches lse or lose. **Note** You must enter **Ctrl+V** and then the question mark or else the help function is invoked.
*	Asterisk	A quantifier that indicates that there are 0, 1 or any number of the previous expression. For example, **lo*se** matches lse, lose, loose, and so on.
+	Plus	A quantifier that indicates that there is at least 1 of the previous expression. For example, **lo+se** matches lose and loose, but not lse.
{*x*} or {*x*,}	Minimum repeat quantifier	Repeat at least *x* times. For example, **ab(xy){2,}z** matches abxyxyz, abxyxyxyz, and so on.
[*abc*]	Character class	Matches any character in the brackets. For example, **[abc]** matches a, b, or c.
[^*abc*]	Negated character class	Matches a single character that is not contained within the brackets. For example, **[^abc]** matches any character other than a, b, or c. **[^A-Z]** matches any single character that is not an uppercase letter.

Table B-2 regex Metacharacters (continued)

Character	Description	Notes
[*a*-*c*]	Character range class	Matches any character in the range. **[a-z]** matches any lowercase letter. You can mix characters and ranges: **[abcq-z]** matches a, b, c, q, r, s, t, u, v, w, x, y, z, and so does [**a-cq-z]**. The dash (-) character is literal only if it is the last or the first character within the brackets: **[abc-]** or **[-abc]**.
""	Quotation marks	Preserves trailing or leading spaces in the string. For example, **" test"** preserves the leading space when it looks for a match.
^	Caret	Specifies the beginning of a line.
****	Escape character	When used with a metacharacter, matches a literal character. For example, **\[** matches the left square bracket.
char	Character	When character is not a metacharacter, matches the literal character.
\r	Carriage return	Matches a carriage return 0x0d.
\n	Newline	Matches a new line 0x0a.
\t	Tab	Matches a tab 0x09.
\f	Formfeed	Matches a form feed 0x0c.
\x*NN*	Escaped hexadecimal number	Matches an ASCII character using hexadecimal (exactly two digits).
NNN	Escaped octal number	Matches an ASCII character as octal (exactly three digits). For example, the character 040 represents a space.

Command Output Paging

On commands such as **help** or**?**, **show**, **show xlate**, or other commands that provide long listings, you can determine if the information displays a screen and pauses, or lets the command run to completion. The **pager** command lets you choose the number of lines to display before the More prompt appears.

When paging is enabled, the following prompt appears:

```
<--- More --->
```

The More prompt uses syntax similar to the UNIX **more** command:

- To view another screen, press the Space bar.
- To view the next line, press the **Enter** key.
- To return to the command line, press the **q** key.

Adding Comments

You can precede a line with a colon (:) to create a comment. However, the comment only appears in the command history buffer and not in the configuration. Therefore, you can view the comment with the **show history** command or by pressing an arrow key to retrieve a previous command, but because the comment is not in the configuration, the **write terminal** command does not display it.

Text Configuration Files

This section describes how to format a text configuration file that you can download to the security appliance, and includes the following topics:

- How Commands Correspond with Lines in the Text File, page B-7
- Command-Specific Configuration Mode Commands, page B-7
- Automatic Text Entries, page B-8
- Line Order, page B-8
- Commands Not Included in the Text Configuration, page B-8
- Passwords, page B-8
- Multiple Security Context Files, page B-8

How Commands Correspond with Lines in the Text File

The text configuration file includes lines that correspond with the commands described in this guide.

In examples, commands are preceded by a CLI prompt. The prompt in the following example is "hostname(config)#":

```
hostname(config)# context a
```

In the text configuration file you are not prompted to enter commands, so the prompt is omitted:

```
context a
```

Command-Specific Configuration Mode Commands

Command-specific configuration mode commands appear indented under the main command when entered at the command line. Your text file lines do not need to be indented, as long as the commands appear directly following the main command. For example, the following unindented text is read the same as indented text:

```
interface gigabitethernet0/0
nameif inside
interface gigabitethernet0/1
    nameif outside
```

Automatic Text Entries

When you download a configuration to the security appliance, the security appliance inserts some lines automatically. For example, the security appliance inserts lines for default settings or for the time the configuration was modified. You do not need to enter these automatic entries when you create your text file.

Line Order

For the most part, commands can be in any order in the file. However, some lines, such as ACEs, are processed in the order they appear, and the order can affect the function of the access list. Other commands might also have order requirements. For example, you must enter the **nameif** command for an interface first because many subsequent commands use the name of the interface. Also, commands in a command-specific configuration mode must directly follow the main command.

Commands Not Included in the Text Configuration

Some commands do not insert lines in the configuration. For example, a runtime command such as **show running-config** does not have a corresponding line in the text file.

Passwords

The login, enable, and user passwords are automatically encrypted before they are stored in the configuration. For example, the encrypted form of the password “cisco” might look like jMorNbK0514fadBh. You can copy the configuration passwords to another security appliance in their encrypted form, but you cannot unencrypt the passwords yourself.

If you enter an unencrypted password in a text file, the security appliance does not automatically encrypt them when you copy the configuration to the security appliance. The security appliance only encrypts them when you save the running configuration from the command line using the **copy running-config startup-config** or **write memory** command.

Multiple Security Context Files

For multiple security contexts, the entire configuration consists of multiple parts:

- The security context configurations
- The system configuration, which identifies basic settings for the security appliance, including a list of contexts
- The admin context, which provides network interfaces for the system configuration

  The system configuration does not include any interfaces or network settings for itself. Rather, when the system needs to access network resources (such as downloading the contexts from the server), it uses a context that is designated as the admin context.

Each context is similar to a single context mode configuration. The system configuration differs from a context configuration in that the system configuration includes system-only commands (such as a list of all contexts) while other typical commands are not present (such as many interface parameters).

Supported Character Sets

The security appliance CLI currently supports UTF-8 encoding only. UTF-8 is the particular encoding scheme for Unicode symbols, and has been designed to be compatible with an ASCII subset of symbols. ASCII characters are represented in UTF-8 as one-byte characters. All other characters are represented in UTF-8 as multi-byte symbols.

The ASCII printable characters (0x20 to 0x7e) are fully supported. The printable ASCII characters are the same as ISO 8859-1. UTF-8 is a superset of ISO 8859-1, so the first 256 characters (0-255) are the same as ISO 8859-1. The security appliance CLI supports up to 255 characters (multi-byte characters) of ISO 8859-1.



APPENDIX C

Addresses, Protocols, and Ports

This appendix provides a quick reference for IP addresses, protocols, and applications. This appendix includes the following sections:

- IPv4 Addresses and Subnet Masks, page C-1
- IPv6 Addresses, page C-5
- Protocols and Applications, page C-11
- TCP and UDP Ports, page C-11
- Local Ports and Protocols, page C-14
- ICMP Types, page C-15

IPv4 Addresses and Subnet Masks

This section describes how to use IPv4 addresses in the security appliance. An IPv4 address is a 32-bit number written in dotted-decimal notation: four 8-bit fields (octets) converted from binary to decimal numbers, separated by dots. The first part of an IP address identifies the network on which the host resides, while the second part identifies the particular host on the given network. The network number field is called the network prefix. All hosts on a given network share the same network prefix but must have a unique host number. In classful IP, the class of the address determines the boundary between the network prefix and the host number.

This section includes the following topics:

- Classes, page C-1
- Private Networks, page C-2
- Subnet Masks, page C-2

Classes

IP host addresses are divided into three different address classes: Class A, Class B, and Class C. Each class fixes the boundary between the network prefix and the host number at a different point within the 32-bit address. Class D addresses are reserved for multicast IP.

- Class A addresses (1.xxx.xxx.xxx through 126.xxx.xxx.xxx) use only the first octet as the network prefix.

- Class B addresses (128.0.xxx.xxx through 191.255.xxx.xxx) use the first two octets as the network prefix.
- Class C addresses (192.0.0.xxx through 223.255.255.xxx) use the first three octets as the network prefix.

Because Class A addresses have 16,777,214 host addresses, and Class B addresses 65,534 hosts, you can use subnet masking to break these huge networks into smaller subnets.

Private Networks

If you need large numbers of addresses on your network, and they do not need to be routed on the Internet, you can use private IP addresses that the Internet Assigned Numbers Authority (IANA) recommends (see RFC 1918). The following address ranges are designated as private networks that should not be advertised:

- 10.0.0.0 through 10.255.255.255
- 172.16.0.0 through 172.31.255.255
- 192.168.0.0 through 192.168.255.255

Subnet Masks

A subnet mask lets you convert a single Class A, B, or C network into multiple networks. With a subnet mask, you can create an extended network prefix that adds bits from the host number to the network prefix. For example, a Class C network prefix always consists of the first three octets of the IP address. But a Class C extended network prefix uses part of the fourth octet as well.

Subnet masking is easy to understand if you use binary notation instead of dotted decimal. The bits in the subnet mask have a one-to-one correspondence with the Internet address:

- The bits are set to 1 if the corresponding bit in the IP address is part of the extended network prefix.
- The bits are set to 0 if the bit is part of the host number.

Example 1: If you have the Class B address 129.10.0.0 and you want to use the entire third octet as part of the extended network prefix instead of the host number, you must specify a subnet mask of 11111111.11111111.11111111.00000000. This subnet mask converts the Class B address into the equivalent of a Class C address, where the host number consists of the last octet only.

Example 2: If you want to use only part of the third octet for the extended network prefix, then you must specify a subnet mask like 11111111.11111111.11111000.00000000, which uses only 5 bits of the third octet for the extended network prefix.

You can write a subnet mask as a dotted-decimal mask or as a */bits* (“slash *bits*”) mask. In Example 1, for a dotted-decimal mask, you convert each binary octet into a decimal number: 255.255.255.0. For a */bits* mask, you add the number of 1s: /24. In Example 2, the decimal number is 255.255.248.0 and the /bits is /21.

You can also supernet multiple Class C networks into a larger network by using part of the third octet for the extended network prefix. For example, 192.168.0.0/20.

This section includes the following topics:

- Determining the Subnet Mask, page C-3
- Determining the Address to Use with the Subnet Mask, page C-3

Determining the Subnet Mask

To determine the subnet mask based on how many hosts you want, see Table C-1.

Table C-1 Hosts, Bits, and Dotted-Decimal Masks

Hosts[1]	/Bits Mask	Dotted-Decimal Mask
16,777,216	/8	255.0.0.0 Class A Network
65,536	/16	255.255.0.0 Class B Network
32,768	/17	255.255.128.0
16,384	/18	255.255.192.0
8192	/19	255.255.224.0
4096	/20	255.255.240.0
2048	/21	255.255.248.0
1024	/22	255.255.252.0
512	/23	255.255.254.0
256	/24	255.255.255.0 Class C Network
128	/25	255.255.255.128
64	/26	255.255.255.192
32	/27	255.255.255.224
16	/28	255.255.255.240
8	/29	255.255.255.248
4	/30	255.255.255.252
Do not use	/31	255.255.255.254
1	/32	255.255.255.255 Single Host Address

1. The first and last number of a subnet are reserved, except for /32, which identifies a single host.

Determining the Address to Use with the Subnet Mask

The following sections describe how to determine the network address to use with a subnet mask for a Class C-size and a Class B-size network. This section includes the following topics:

- Class C-Size Network Address, page C-3
- Class B-Size Network Address, page C-4

Class C-Size Network Address

For a network between 2 and 254 hosts, the fourth octet falls on a multiple of the number of host addresses, starting with 0. For example, the 8-host subnets (/29) of 192.168.0.x are as follows:

Subnet with Mask /29 (255.255.255.248)	Address Range[1]
192.168.0.0	192.168.0.0 to 192.168.0.7
192.168.0.8	192.168.0.8 to 192.168.0.15

Subnet with Mask /29 (255.255.255.248)	Address Range[1]
192.168.0.16	192.168.0.16 to 192.168.0.31
…	…
192.168.0.248	192.168.0.248 to 192.168.0.255

1. The first and last address of a subnet are reserved. In the first subnet example, you cannot use 192.168.0.0 or 192.168.0.7.

Class B-Size Network Address

To determine the network address to use with the subnet mask for a network with between 254 and 65,534 hosts, you need to determine the value of the third octet for each possible extended network prefix. For example, you might want to subnet an address like 10.1.*x*.0, where the first two octets are fixed because they are used in the extended network prefix, and the fourth octet is 0 because all bits are used for the host number.

To determine the value of the third octet, follow these steps:

Step 1 Calculate how many subnets you can make from the network by dividing 65,536 (the total number of addresses using the third and fourth octet) by the number of host addresses you want.

For example, 65,536 divided by 4096 hosts equals 16.

Therefore, there are 16 subnets of 4096 addresses each in a Class B-size network.

Step 2 Determine the multiple of the third octet value by dividing 256 (the number of values for the third octet) by the number of subnets:

In this example, 256/16 = 16.

The third octet falls on a multiple of 16, starting with 0.

Therefore, the 16 subnets of the network 10.1 are as follows:

Subnet with Mask /20 (255.255.240.0)	Address Range[1]
10.1.0.0	10.1.0.0 to 10.1.15.255
10.1.16.0	10.1.16.0 to 10.1.31.255
10.1.32.0	10.1.32.0 to 10.1.47.255
…	…
10.1.240.0	10.1.240.0 to 10.1.255.255

1. The first and last address of a subnet are reserved. In the first subnet example, you cannot use 10.1.0.0 or 10.1.15.255.

IPv6 Addresses

IPv6 is the next generation of the Internet Protocol after IPv4. It provides an expanded address space, a simplified header format, improved support for extensions and options, flow labeling capability, and authentication and privacy capabilities. IPv6 is described in RFC 2460. The IPv6 addressing architecture is described in RFC 3513.

This section describes the IPv6 address format and architecture and includes the following topics:

- IPv6 Address Format, page C-5
- IPv6 Address Types, page C-6
- IPv6 Address Prefixes, page C-10

Note This section describes the IPv6 address format, the types, and prefixes. For information about configuring the security appliance to use IPv6, see Chapter 8, "Configuring Interface Parameters."

IPv6 Address Format

IPv6 addresses are represented as a series of eight 16-bit hexadecimal fields separated by colons (:) in the format: x:x:x:x:x:x:x:x. The following are two examples of IPv6 addresses:

- 2001:0DB8:7654:3210:FEDC:BA98:7654:3210
- 2001:0DB8:0000:0000:0008:0800:200C:417A

Note The hexadecimal letters in IPv6 addresses are not case-sensitive.

It is not necessary to include the leading zeros in an individual field of the address. But each field must contain at least one digit. So the example address 2001:0DB8:0000:0000:0008:0800:200C:417A can be shortened to 2001:0DB8:0:0:8:800:200C:417A by removing the leading zeros from the third through sixth fields from the left. The fields that contained all zeros (the third and fourth fields from the left) were shortened to a single zero. The fifth field from the left had the three leading zeros removed, leaving a single 8 in that field, and the sixth field from the left had the one leading zero removed, leaving 800 in that field.

It is common for IPv6 addresses to contain several consecutive hexadecimal fields of zeros. You can use two colons (::) to compress consecutive fields of zeros at the beginning, middle, or end of an IPv6 address (the colons represent the successive hexadecimal fields of zeros). Table C-2 shows several examples of address compression for different types of IPv6 address.

Table C-2 IPv6 Address Compression Examples

Address Type	Standard Form	Compressed Form
Unicast	2001:0DB8:0:0:0:BA98:0:3210	2001:0DB8::BA98:0:3210
Multicast	FF01:0:0:0:0:0:0:101	FF01::101
Loopback	0:0:0:0:0:0:0:1	::1
Unspecified	0:0:0:0:0:0:0:0	::

Note Two colons (::) can be used only once in an IPv6 address to represent successive fields of zeros.

An alternative form of the IPv6 format is often used when dealing with an environment that contains both IPv4 and IPv6 addresses. This alternative has the format x:x:x:x:x:x:y.y.y.y, where x represent the hexadecimal values for the six high-order parts of the IPv6 address and y represent decimal values for the 32-bit IPv4 part of the address (which takes the place of the remaining two 16-bit parts of the IPv6 address). For example, the IPv4 address 192.168.1.1 could be represented as the IPv6 address 0:0:0:0:0:0:FFFF:192.168.1.1, or ::FFFF:192.168.1.1.

IPv6 Address Types

The following are the three main types of IPv6 addresses:

- **Unicast**—A unicast address is an identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that address. An interface may have more than one unicast address assigned to it.
- **Multicast**—A multicast address is an identifier for a set of interfaces. A packet sent to a multicast address is delivered to all addresses identified by that address.
- **Anycast**—An anycast address is an identifier for a set of interfaces. Unlike a multicast address, a packet sent to an anycast address is only delivered to the "nearest" interface, as determined by the measure of distances for the routing protocol.

Note There are no broadcast addresses in IPv6. Multicast addresses provide the broadcast functionality.

This section includes the following topics:

- Unicast Addresses, page C-6
- Multicast Address, page C-8
- Anycast Address, page C-9
- Required Addresses, page C-10

Unicast Addresses

This section describes IPv6 unicast addresses. Unicast addresses identify an interface on a network node.

This section includes the following topics:

- Global Address, page C-7
- Site-Local Address, page C-7
- Link-Local Address, page C-7
- IPv4-Compatible IPv6 Addresses, page C-7
- Unspecified Address, page C-8
- Loopback Address, page C-8
- Interface Identifiers, page C-8

Global Address

The general format of an IPv6 global unicast address is a global routing prefix followed by a subnet ID followed by an interface ID. The global routing prefix can be any prefix not reserved by another IPv6 address type (see IPv6 Address Prefixes, page C-10, for information about the IPv6 address type prefixes).

All global unicast addresses, other than those that start with binary 000, have a 64-bit interface ID in the Modified EUI-64 format. See Interface Identifiers, page C-8, for more information about the Modified EUI-64 format for interface identifiers.

Global unicast address that start with the binary 000 do not have any constraints on the size or structure of the interface ID portion of the address. One example of this type of address is an IPv6 address with an embedded IPv4 address (see IPv4-Compatible IPv6 Addresses, page C-7).

Site-Local Address

Site-local addresses are used for addressing within a site. They can be use to address an entire site without using a globally unique prefix. Site-local addresses have the prefix FEC0::/10, followed by a 54-bit subnet ID, and end with a 64-bit interface ID in the modified EUI-64 format.

Site-local Routers do not forward any packets that have a site-local address for a source or destination outside of the site. Therefore, site-local addresses can be considered private addresses.

Link-Local Address

All interfaces are required to have at least one link-local address. You can configure multiple IPv6 addresses per interfaces, but only one link-local address.

A link-local address is an IPv6 unicast address that can be automatically configured on any interface using the link-local prefix FE80::/10 and the interface identifier in modified EUI-64 format. Link-local addresses are used in the neighbor discovery protocol and the stateless autoconfiguration process. Nodes with a link-local address can communicate; they do not need a site-local or globally unique address to communicate.

Routers do not forward any packets that have a link-local address for a source or destination. Therefore, link-local addresses can be considered private addresses.

IPv4-Compatible IPv6 Addresses

There are two types of IPv6 addresses that can contain IPv4 addresses.

The first type is the “IPv4-compatibly IPv6 address.” The IPv6 transition mechanisms include a technique for hosts and routers to dynamically tunnel IPv6 packets over IPv4 routing infrastructure. IPv6 nodes that use this technique are assigned special IPv6 unicast addresses that carry a global IPv4 address in the low-order 32 bits. This type of address is termed an “IPv4-compatible IPv6 address” and has the format ::y.y.y.y, where y.y.y.y is an IPv4 unicast address.



Note The IPv4 address used in the “IPv4-compatible IPv6 address” must be a globally-unique IPv4 unicast address.

The second type of IPv6 address which holds an embedded IPv4 address is called the “IPv4-mapped IPv6 address.” This address type is used to represent the addresses of IPv4 nodes as IPv6 addresses. This type of address has the format ::FFFF:y.y.y.y, where y.y.y.y is an IPv4 unicast address.

Unspecified Address

The unspecified address, 0:0:0:0:0:0:0:0, indicates the absence of an IPv6 address. For example, a newly initialized node on an IPv6 network may use the unspecified address as the source address in its packets until it receives its IPv6 address.

Note The IPv6 unspecified address cannot be assigned to an interface. The unspecified IPv6 addresses must not be used as destination addresses in IPv6 packets or the IPv6 routing header.

Loopback Address

The loopback address, 0:0:0:0:0:0:0:1, may be used by a node to send an IPv6 packet to itself. The loopback address in IPv6 functions the same as the loopback address in IPv4 (127.0.0.1).

Note The IPv6 loopback address cannot be assigned to a physical interface. A packet that has the IPv6 loopback address as its source or destination address must remain within the node that created the packet. IPv6 routers do not forward packets that have the IPv6 loopback address as their source or destination address.

Interface Identifiers

Interface identifiers in IPv6 unicast addresses are used to identify the interfaces on a link. They need to be unique within a subnet prefix. In many cases, the interface identifier is derived from the interface link-layer address. The same interface identifier may be used on multiple interfaces of a single node, as long as those interfaces are attached to different subnets.

For all unicast addresses, except those that start with the binary 000, the interface identifier is required to be 64 bits long and to be constructed in the Modified EUI-64 format. The Modified EUI-64 format is created from the 48-bit MAC address by inverting the universal/local bit in the address and by inserting the hexadecimal number FFFE between the upper three bytes and lower three bytes of the of the MAC address.

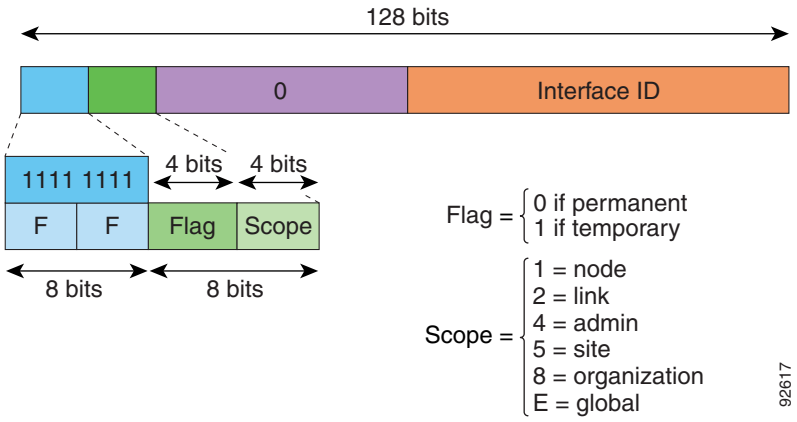
For example, and interface with the MAC address of 00E0.b601.3B7A would have a 64-bit interface ID of 02E0:B6FF:FE01:3B7A.

Multicast Address

An IPv6 multicast address is an identifier for a group of interfaces, typically on different nodes. A packet sent to a multicast address is delivered to all interfaces identified by the multicast address. An interface may belong to any number of multicast groups.

An IPv6 multicast address has a prefix of FF00::/8 (1111 1111). The octet following the prefix defines the type and scope of the multicast address. A permanently assigned (“well known”) multicast address has a flag parameter equal to 0; a temporary (“transient”) multicast address has a flag parameter equal to 1. A multicast address that has the scope of a node, link, site, or organization, or a global scope has a scope parameter of 1, 2, 5, 8, or E, respectively. For example, a multicast address with the prefix FF02::/16 is a permanent multicast address with a link scope. Figure C-1 shows the format of the IPv6 multicast address.

Figure C-1 IPv6 Multicast Address Format



IPv6 nodes (hosts and routers) are required to join the following multicast groups:

- The All Nodes multicast addresses:
  - FF01:: (interface-local)
  - FF02:: (link-local)
- The Solicited-Node Address for each IPv6 unicast and anycast address on the node: FF02:0:0:0:0:1:FFXX:XXXX/104, where XX:XXXX is the low-order 24-bits of the unicast or anycast address.

Note Solicited-Node addresses are used in Neighbor Solicitation messages.

IPv6 routers are required to join the following multicast groups:

- FF01::2 (interface-local)
- FF02::2 (link-local)
- FF05::2 (site-local)

Multicast address should not be used as source addresses in IPv6 packets.

Note There are no broadcast addresses in IPv6. IPv6 multicast addresses are used instead of broadcast addresses.

Anycast Address

The IPv6 anycast address is a unicast address that is assigned to more than one interface (typically belonging to different nodes). A packet that is routed to an anycast address is routed to the nearest interface having that address, the nearness being determined by the routing protocol in effect.

Anycast addresses are allocated from the unicast address space. An anycast address is simply a unicast address that has been assigned to more than one interface, and the interfaces must be configured to recognize the address as an anycast address.

The following restrictions apply to anycast addresses:

- An anycast address cannot be used as the source address for an IPv6 packet.

- An anycast address cannot be assigned to an IPv6 host; it can only be assigned to an IPv6 router.

Note Anycast addresses are not supported on the security appliance.

Required Addresses

IPv6 hosts must, at a minimum, be configured with the following addresses (either automatically or manually):

- A link-local address for each interface.
- The loopback address.
- The All-Nodes multicast addresses
- A Solicited-Node multicast address for each unicast or anycast address.

IPv6 routers must, at a minimum, be configured with the following addresses (either automatically or manually):

- The required host addresses.
- The Subnet-Router anycast addresses for all interfaces for which it is configured to act as a router.
- The All-Routers multicast addresses.

IPv6 Address Prefixes

An IPv6 address prefix, in the format ipv6-prefix/prefix-length, can be used to represent bit-wise contiguous blocks of the entire address space. The IPv6-prefix must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. The prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). For example, 2001:0DB8:8086:6502::/32 is a valid IPv6 prefix.

The IPv6 prefix identifies the type of IPv6 address. Table C-3 shows the prefixes for each IPv6 address type.

Table C-3 IPv6 Address Type Prefixes

Address Type	Binary Prefix	IPv6 Notation
Unspecified	000...0 (128 bits)	::/128
Loopback	000...1 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-Local (unicast)	1111111010	FE80::/10
Site-Local (unicast)	1111111111	FEC0::/10
Global (unicast)	All other addresses.	
Anycast	Taken from the unicast address space.	

Protocols and Applications

Table C-4 lists the protocol literal values and port numbers; either can be entered in security appliance commands.

Table C-4 Protocol Literal Values

Literal	Value	Description
ah	51	Authentication Header for IPv6, RFC 1826.
eigrp	88	Enhanced Interior Gateway Routing Protocol.
esp	50	Encapsulated Security Payload for IPv6, RFC 1827.
gre	47	Generic Routing Encapsulation.
icmp	1	Internet Control Message Protocol, RFC 792.
icmp6	58	Internet Control Message Protocol for IPv6, RFC 2463.
igmp	2	Internet Group Management Protocol, RFC 1112.
igrp	9	Interior Gateway Routing Protocol.
ip	0	Internet Protocol.
ipinip	4	IP-in-IP encapsulation.
ipsec	50	IP Security. Entering the ipsec protocol literal is equivalent to entering the esp protocol literal.
nos	94	Network Operating System (Novell's NetWare).
ospf	89	Open Shortest Path First routing protocol, RFC 1247.
pcp	108	Payload Compression Protocol.
pim	103	Protocol Independent Multicast.
pptp	47	Point-to-Point Tunneling Protocol. Entering the pptp protocol literal is equivalent to entering the gre protocol literal.
snp	109	Sitara Networks Protocol.
tcp	6	Transmission Control Protocol, RFC 793.
udp	17	User Datagram Protocol, RFC 768.

Protocol numbers can be viewed online at the IANA website:

http://www.iana.org/assignments/protocol-numbers

TCP and UDP Ports

Table C-5 lists the literal values and port numbers; either can be entered in security appliance commands. See the following caveats:

- The security appliance uses port 1521 for SQL*Net. This is the default port used by Oracle for SQL*Net. This value, however, does not agree with IANA port assignments.

- The security appliance listens for RADIUS on ports 1645 and 1646. If your RADIUS server uses the standard ports 1812 and 1813, you can configure the security appliance to listen to those ports using the **authentication-port** and **accounting-port** commands.
- To assign a port for DNS access, use the **domain** literal value, not **dns**. If you use **dns**, the security appliance assumes you meant to use the **dnsix** literal value.

Port numbers can be viewed online at the IANA website:

http://www.iana.org/assignments/port-numbers

Table C-5 Port Literal Values

Literal	TCP or UDP?	Value	Description
aol	TCP	5190	America Online
bgp	TCP	179	Border Gateway Protocol, RFC 1163
biff	UDP	512	Used by mail system to notify users that new mail is received
bootpc	UDP	68	Bootstrap Protocol Client
bootps	UDP	67	Bootstrap Protocol Server
chargen	TCP	19	Character Generator
citrix-ica	TCP	1494	Citrix Independent Computing Architecture (ICA) protocol
cmd	TCP	514	Similar to **exec** except that **cmd** has automatic authentication
ctiqbe	TCP	2748	Computer Telephony Interface Quick Buffer Encoding
daytime	TCP	13	Day time, RFC 867
discard	TCP, UDP	9	Discard
domain	TCP, UDP	53	DNS
dnsix	UDP	195	DNSIX Session Management Module Audit Redirector
echo	TCP, UDP	7	Echo
exec	TCP	512	Remote process execution
finger	TCP	79	Finger
ftp	TCP	21	File Transfer Protocol (control port)
ftp-data	TCP	20	File Transfer Protocol (data port)
gopher	TCP	70	Gopher
https	TCP	443	HTTP over SSL
h323	TCP	1720	H.323 call signalling
hostname	TCP	101	NIC Host Name Server
ident	TCP	113	Ident authentication service
imap4	TCP	143	Internet Message Access Protocol, version 4
irc	TCP	194	Internet Relay Chat protocol

Table C-5 Port Literal Values (continued)

Literal	TCP or UDP?	Value	Description
isakmp	UDP	500	Internet Security Association and Key Management Protocol
kerberos	TCP, UDP	750	Kerberos
klogin	TCP	543	KLOGIN
kshell	TCP	544	Korn Shell
ldap	TCP	389	Lightweight Directory Access Protocol
ldaps	TCP	636	Lightweight Directory Access Protocol (SSL)
lpd	TCP	515	Line Printer Daemon - printer spooler
login	TCP	513	Remote login
lotusnotes	TCP	1352	IBM Lotus Notes
mobile-ip	UDP	434	MobileIP-Agent
nameserver	UDP	42	Host Name Server
netbios-ns	UDP	137	NetBIOS Name Service
netbios-dgm	UDP	138	NetBIOS Datagram Service
netbios-ssn	TCP	139	NetBIOS Session Service
nntp	TCP	119	Network News Transfer Protocol
ntp	UDP	123	Network Time Protocol
pcanywhere-status	UDP	5632	pcAnywhere status
pcanywhere-data	TCP	5631	pcAnywhere data
pim-auto-rp	TCP, UDP	496	Protocol Independent Multicast, reverse path flooding, dense mode
pop2	TCP	109	Post Office Protocol - Version 2
pop3	TCP	110	Post Office Protocol - Version 3
pptp	TCP	1723	Point-to-Point Tunneling Protocol
radius	UDP	1645	Remote Authentication Dial-In User Service
radius-acct	UDP	1646	Remote Authentication Dial-In User Service (accounting)
rip	UDP	520	Routing Information Protocol
secureid-udp	UDP	5510	SecureID over UDP
smtp	TCP	25	Simple Mail Transport Protocol
snmp	UDP	161	Simple Network Management Protocol
snmptrap	UDP	162	Simple Network Management Protocol - Trap
sqlnet	TCP	1521	Structured Query Language Network
ssh	TCP	22	Secure Shell
sunrpc (rpc)	TCP, UDP	111	Sun Remote Procedure Call
syslog	UDP	514	System Log

Table C-5 Port Literal Values (continued)

Literal	TCP or UDP?	Value	Description
tacacs	TCP, UDP	49	Terminal Access Controller Access Control System Plus
talk	TCP, UDP	517	Talk
telnet	TCP	23	RFC 854 Telnet
tftp	UDP	69	Trivial File Transfer Protocol
time	UDP	37	Time
uucp	TCP	540	UNIX-to-UNIX Copy Program
who	UDP	513	Who
whois	TCP	43	Who Is
www	TCP	80	World Wide Web
xdmcp	UDP	177	X Display Manager Control Protocol

Local Ports and Protocols

Table C-6 lists the protocols, TCP ports, and UDP ports that the security appliance may open to process traffic destined to the security appliance. Unless you enable the features and services listed in Table C-6, the security appliance does *not* open any local protocols or any TCP or UDP ports. You must configure a feature or service for the security appliance to open the default listening protocol or port. In many cases you can configure ports other than the default port when you enable a feature or service.

Table C-6 Protocols and Ports Opened by Features and Services

Feature or Service	Protocol	Port Number	Comments
DHCP	UDP	67,68	—
Failover Control	108	N/A	—
HTTP	TCP	80	—
HTTPS	TCP	443	—
ICMP	1	N/A	—
IGMP	2	N/A	Protocol only open on destination IP address 224.0.0.1
ISAKMP/IKE	UDP	500	Configurable.
IPSec (ESP)	50	N/A	—
IPSec over UDP (NAT-T)	UDP	4500	—
IPSec over UDP (Cisco VPN 3000 Series compatible)	UDP	10000	Configurable.
IPSec over TCP (CTCP)	TCP	—	No default port is used. You must specify the port number when configuring IPSec over TCP.

Table C-6 Protocols and Ports Opened by Features and Services (continued)

Feature or Service	Protocol	Port Number	Comments
NTP	UDP	123	—
OSPF	89	N/A	Protocol only open on destination IP address 224.0.0.5 and 224.0.0.6
PIM	103	N/A	Protocol only open on destination IP address 224.0.0.13
RIP	UDP	520	—
RIPv2	UDP	520	Port only open on destination IP address 224.0.0.9
SNMP	UDP	161	Configurable.
SSH	TCP	22	—
Stateful Update	105	N/A	—
Telnet	TCP	23	—
VPN Load Balancing	UDP	9023	Configurable.
VPN Individual User Authentication Proxy	UDP	1645, 1646	Port accessible only over VPN tunnel.

ICMP Types

Table C-7 lists the ICMP type numbers and names that you can enter in security appliance commands:

Table C-7 ICMP Types

ICMP Number	ICMP Name
0	echo-reply
3	unreachable
4	source-quench
5	redirect
6	alternate-address
8	echo
9	router-advertisement
10	router-solicitation
11	time-exceeded
12	parameter-problem
13	timestamp-request
14	timestamp-reply
15	information-request
16	information-reply
17	mask-request

Table C-7 ICMP Types (continued)

ICMP Number	ICMP Name
18	mask-reply
31	conversion-error
32	mobile-redirect



APPENDIX D

Configuring an External Server for Authorization and Authentication

This appendix describes how to configure an external LDAP, RADIUS, or TACACS+ server to support AAA on the security appliance. Before you configure the security appliance to use an external server, you must configure the server with the correct security appliance authorization attributes and, from a subset of these attributes, assign specific permissions to individual users.

This appendix includes the following sections:

- Understanding Policy Enforcement of Permissions and Attributes, page D-2
- Configuring an External LDAP Server, page D-3
- Configuring an External RADIUS Server, page D-27
- Configuring an External TACACS+ Server, page D-35

Understanding Policy Enforcement of Permissions and Attributes

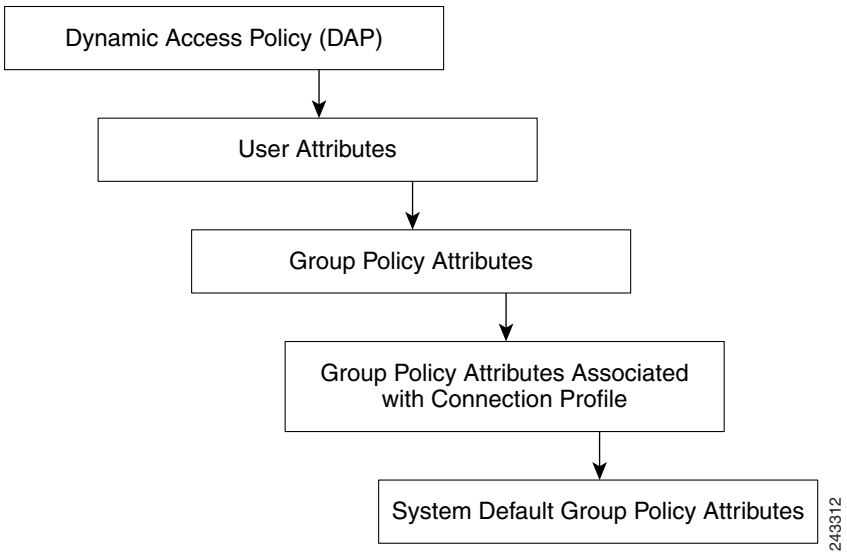
The security appliance supports several methods of applying user authorization attributes (also called user entitlements or permissions) to VPN connections. You can configure the security appliance to obtain user attributes from a Dynamic Access Policy (DAP) on the security appliance, from an external authentication and/or authorization AAA server (RADIUS or LDAP), from a group policy on the security appliance, or from all three.

If the security appliance receives attributes from all sources, the attributes are evaluated, merged, and applied to the user policy. If there are conflicts between attributes coming from the DAP, the AAA server, or the group policy, those attributes obtained from the DAP always take precedence.

The security appliance applies attributes in the following order (also illustrated in Figure D-1:

1. DAP attributes on the security appliance—Introduced in Version 8.0, take precedence over all others. If you set a bookmark/URL list in DAP, it overrides a bookmark/URL list set in the group policy.
2. User attributes on the AAA server—The server returns these after successful user authentication and/or authorization. Do not confuse these with attributes that are set for individual users in the local AAA database on the security appliance (User Accounts in ASDM).
3. Group policy configured on the security appliance—If a RADIUS server returns the value of the RADIUS CLASS attribute IETF-Class-25 (OU=<group-policy>) for the user, the security appliance places the user in the group policy of the same name and enforces any attributes in the group policy that are not returned by the server. For LDAP servers, any attribute name can be used to set the group policy for the session. The LDAP attribute map you configure on the security appliance maps the LDAP attribute to the Cisco attribute IETF-Radius-Class.
4. Group policy assigned by the Connection Profile (called *tunnel-group* in CLI)—The Connection Profile has the preliminary settings for the connection, and includes a default group policy applied to the user before authentication. All users connecting to the security appliance initially belong to this group which provides any attributes that are missing from the DAP, user attributes returned by the server, or the group policy assigned to the user.
5. Default group policy assigned by the security appliance (DfltGrpPolicy)—System default attributes provide any values that are missing from the DAP, user attributes, group policy, or connection profile.

Figure D-1 Policy Enforcement Flow



Configuring an External LDAP Server

The VPN 3000 Concentrator and the ASA/PIX 7.0 required a Cisco LDAP schema for authorization operations. Beginning with Version 7.1.x, the security appliance performs authentication *and* authorization, using the native LDAP schema, and the Cisco schema is no longer needed.

You configure authorization (permission policy) using an LDAP attribute map. For examples, see Active Directory/LDAP VPN Remote Access Authorization Use Cases, page D-14.

This section describes the structure, schema, and attributes of an LDAP server. It includes the following topics:

- Organizing the Security Appliance for LDAP Operations, page D-3
- Defining the Security Appliance LDAP Configuration, page D-5
- Active Directory/LDAP VPN Remote Access Authorization Use Cases, page D-14

The specific steps of these processes vary, depending on which type of LDAP server you are using.

Note For more information on the LDAP protocol, see RFCs 1777, 2251, and 2849.

Organizing the Security Appliance for LDAP Operations

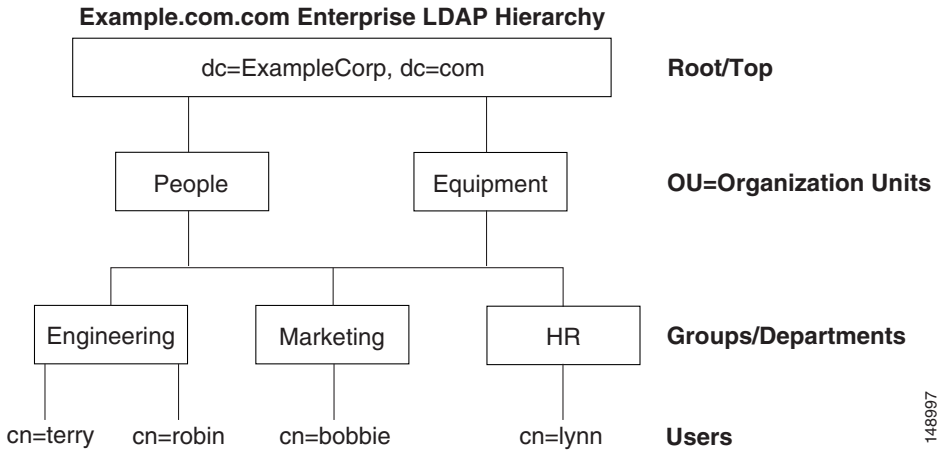
This section describes how to perform searches within the LDAP hierarchy and authenticated binding to the LDAP server on the security appliance. It includes the following topics:

- Searching the Hierarchy, page D-4
- Binding the Security Appliance to the LDAP Server, page D-5
- Login DN Example for Active Directory, page D-5

Your LDAP configuration should reflect the logical hierarchy of your organization. For example, suppose an employee at your company, Example Corporation, is named Terry. Terry works in the Engineering group. Your LDAP hierarchy could have one or many levels. You might decide to set up a shallow, single-level hierarchy in which Terry is considered a member of Example Corporation. Or, you could set up a multi-level hierarchy in which Terry is considered to be a member of the department Engineering, which is a member of an organizational unit called People, which is itself a member of Example Corporation. See Figure D-2 for an example of this multi-level hierarchy.

A multi-level hierarchy has more granularity, but a single level hierarchy is quicker to search.

Figure D-2 A Multi-Level LDAP Hierarchy



Searching the Hierarchy

The security appliance lets you tailor the search within the LDAP hierarchy. You configure the following three fields on the security appliance to define where in the LDAP hierarchy your search begins, the extent, and the type of information it is looking for. Together these fields allow you to limit the search of the hierarchy to only the part of the tree that contains the user permissions.

- LDAP Base DN defines where in the LDAP hierarchy the server should begin searching for user information when it receives an authorization request from the security appliance.
- Search Scope defines the extent of the search in the LDAP hierarchy. The search proceeds this many levels in the hierarchy below the LDAP Base DN. You can choose to have the server search only the level immediately below, or it can search the entire subtree. A single level search is quicker, but a subtree search is more extensive.
- Naming Attribute(s) defines the RDN that uniquely identifies an entry in the LDAP server. Common naming attributes can include cn (Common Name), sAMAccountName, and userPrincipalName.

Figure D-2 shows a possible LDAP hierarchy for Example Corporation. Given this hierarchy, you could define your search in different ways. Table D-1 shows two possible search configurations.

In the first example configuration, when Terry establishes the IPSec tunnel with LDAP authorization required, the security appliance sends a search request to the LDAP server indicating it should search for Terry in the Engineering group. This search is quick.

In the second example configuration, the security appliance sends a search request indicating the server should search for Terry within Example Corporation. This search takes longer.

Table D-1 Example Search Configurations

#	LDAP Base DN	Search Scope	Naming Attribute	Result
1	group= Engineering,ou=People,dc=ExampleCorporation, dc=com	One Level	cn=Terry	Quicker search
2	dc=ExampleCorporation,dc=com	Subtree	cn=Terry	Longer search

Binding the Security Appliance to the LDAP Server

Some LDAP servers (including the Microsoft Active Directory server) require the security appliance to establish a handshake via authenticated binding before they accept requests for any other LDAP operations. The security appliance identifies itself for authenticated binding by attaching a Login DN field to the user authentication request. The Login DN field defines the authentication characteristics of the security appliance; these characteristics should correspond to those of a user with administrative privileges. An example Login DN field could be: cn=Administrator, cn=users, ou=people, dc=example, dc=com.

Note As an LDAP client, the security appliance does not support sending anonymous binds or requests.

Login DN Example for Active Directory

The Login DN is a username on the LDAP server that the security appliance uses to establish a trust between itself (the LDAP client) and the LDAP server during the Bind exchange, before a user search can take place.

For VPN authentication/authorization operations, and beginning with version 8.0.4 for retrieval of AD Groups, (which are read operations only when password-management changes are not required), the you can use the Login DN with fewer privileges. For example, the Login DN can be a user who is a memberOf the Domain Users group.

For VPN password-management changes, the Login DN must have Account Operators privileges.

In either of these cases, Super-user level privileges are not required for the Login/Bind DN. Refer to your LDAP Administrator guide for specific Login DN requirements.

Defining the Security Appliance LDAP Configuration

This section describes how to define the LDAP AV-pair attribute syntax. It includes the following topics:

- Supported Cisco Attributes for LDAP Authorization, page D-6
- Cisco-AV-Pair Attribute Syntax, page D-12

Note The security appliance enforces the LDAP attributes based on attribute name, not numeric ID. RADIUS attributes, on the other hand, are enforced by numeric ID, not by name.

Authorization refers to the process of enforcing permissions or attributes. An LDAP server defined as an authentication or authorization server will enforce permissions or attributes if they are configured.

For software Version 7.0, LDAP attributes include the cVPN3000 prefix. For Version 7.1 and later, this prefix was removed.

Supported Cisco Attributes for LDAP Authorization

This section provides a complete list of attributes (Table D-2) for the ASA 5500, VPN 3000, and PIX 500 series security appliances. The table includes attribute support information for the VPN 3000 and PIX 500 series to assist you configure networks with a mixture of these security appliances.

Table D-2 Security Appliance Supported Cisco Attributes for LDAP Authorization

Attribute Name/	VPN 3000	ASA	PIX	Syntax/ Type	Single or Multi-Valued	Possible Values
Access-Hours	Y	Y	Y	String	Single	Name of the time-range (for example, Business-Hours)
Allow-Network-Extension- Mode	Y	Y	Y	Boolean	Single	0 = Disabled 1 = Enabled
Authenticated-User-Idle- Timeout	Y	Y	Y	Integer	Single	1 - 35791394 minutes
Authorization-Required	Y			Integer	Single	0 = No 1 = Yes
Authorization-Type	Y			Integer	Single	0 = None 1 = RADIUS 2 = LDAP
Auth-Service-Type						
Banner1	Y	Y	Y	String	Single	Banner string
Banner2	Y	Y	Y	String	Single	Banner string
Cisco-AV-Pair	Y	Y	Y	String	Multi	An octet string in the following format: [Prefix] [Action] [Protocol] [Source] [Source Wildcard Mask] [Destination] [Destination Wildcard Mask] [Established] [Log] [Operator] [Port] For more information, see "Cisco-AV-Pair Attribute Syntax."
Cisco-IP-Phone-Bypass	Y	Y	Y	Integer	Single	0 = Disabled 1 = Enabled
Cisco-LEAP-Bypass	Y	Y	Y	Integer	Single	0 = Disabled 1 = Enabled
Client-Intercept-DHCP-Configure-Msg	Y	Y	Y	Boolean	Single	0 = Disabled 1 = Enabled
Client-Type-Version-Limiting	Y	Y	Y	String	Single	IPSec VPN client version number string
Confidence-Interval	Y	Y	Y	Integer	Single	10 - 300 seconds

Table D-2 Security Appliance Supported Cisco Attributes for LDAP Authorization (continued)

Attribute Name/	VPN 3000	ASA	PIX	Syntax/ Type	Single or Multi-Valued	Possible Values
DHCP-Network-Scope	Y	Y	Y	String	Single	IP address
DN-Field	Y	Y	Y	String	Single	Possible values: UID, OU, O, CN, L, SP, C, EA, T, N, GN, SN, I, GENQ, DNQ, SER, use-entire-name.
Firewall-ACL-In		Y	Y	String	Single	Access list ID
Firewall-ACL-Out		Y	Y	String	Single	Access list ID
IE-Proxy-Bypass-Local				Boolean	Single	0=Disabled 1=Enabled
IE-Proxy-Exception-List				String	Single	A list of DNS domains. Entries must be separated by the new line character sequence (\n).
IE-Proxy-Method	Y	Y	Y	Integer	Single	1 = Do not modify proxy settings 2 = Do not use proxy 3 = Auto detect 4 = Use security appliance setting
IE-Proxy-Server	Y	Y	Y	Integer	Single	IP Address
IETF-Radius-Class	Y	Y	Y		Single	Sets the group policy for the remote access VPN session
IETF-Radius-Filter-Id	Y	Y	Y	String	Single	access list name that is defined on the security appliance
IETF-Radius-Framed-IP-Address	Y	Y	Y	String	Single	An IP address
IETF-Radius-Framed-IP-Netmask	Y	Y	Y	String	Single	An IP address mask
IETF-Radius-Idle-Timeout	Y	Y	Y	Integer	Single	minutes
IETF-Radius-Service-Type	Y	Y	Y	Integer	Single	
IETF-Radius-Session-Timeout	Y	Y	Y	Integer	Single	
IKE-Keep-Alives	Y	Y	Y	Boolean	Single	0 = Disabled 1 = Enabled
IPSec-Allow-Passwd-Store	Y	Y	Y	Boolean	Single	0 = Disabled 1 = Enabled
IPSec-Authentication	Y	Y	Y	Integer	Single	0 = None 1 = RADIUS 2 = LDAP (authorization only) 3 = NT Domain 4 = SDI (RSA) 5 = Internal 6 = RADIUS with Expiry 7 = Kerberos/Active Directory
IPSec-Auth-On-Rekey	Y	Y	Y	Boolean	Single	0 = Disabled 1 = Enabled
IPSec-Backup-Server-List	Y	Y	Y	String	Single	Server Addresses (space delimited)

Table D-2 Security Appliance Supported Cisco Attributes for LDAP Authorization (continued)

Attribute Name/	VPN 3000	ASA	PIX	Syntax/ Type	Single or Multi-Valued	Possible Values
IPSec-Backup-Servers	Y	Y	Y	String	Single	1 = Use Client-Configured list 2 = Disabled and clear client list 3 = Use Backup Server list
IPSec-Client-Firewall-Filter- Name	Y			String	Single	Specifies the name of the filter to be pushed to the client as firewall policy.
IPSec-Client-Firewall-Filter-Optional	Y	Y	Y	Integer	Single	0 = Required 1 = Optional
IPSec-Default-Domain	Y	Y	Y	String	Single	Specifies the single default domain name to send to the client (1 - 255 characters).
IPSec-IKE-Peer-ID-Check	Y	Y	Y	Integer	Single	1 = Required 2 = If supported by peer certificate 3 = Do not check
IPSec-IP-Compression	Y	Y	Y	Integer	Single	0 = Disabled 1 = Enabled
IPSec-Mode-Config	Y	Y	Y	Boolean	Single	0 = Disabled 1 = Enabled
IPSec-Over-UDP	Y	Y	Y	Boolean	Single	0 = Disabled 1 = Enabled
IPSec-Over-UDP-Port	Y	Y	Y	Integer	Single	4001 - 49151; default = 10000
IPSec-Required-Client-Firewall-Capability	Y	Y	Y	Integer	Single	0 = None 1 = Policy defined by remote FW Are-You-There (AYT) 2 = Policy pushed CPP 4 = Policy from server
IPSec-Sec-Association	Y			String	Single	Name of the security association
IPSec-Split-DNS-Names	Y	Y	Y	String	Single	Specifies the list of secondary domain names to send to the client (1 - 255 characters).
IPSec-Split-Tunneling-Policy	Y	Y	Y	Integer	Single	0 = Tunnel everything 1 = Split tunneling 2 = Local LAN permitted
IPSec-Split-Tunnel-List	Y	Y	Y	String	Single	Specifies the name of the network or access list that describes the split tunnel inclusion list.
IPSec-Tunnel-Type	Y	Y	Y	Integer	Single	1 = LAN-to-LAN 2 = Remote access
IPSec-User-Group-Lock	Y			Boolean	Single	0 = Disabled 1 = Enabled

Table D-2 Security Appliance Supported Cisco Attributes for LDAP Authorization (continued)

Attribute Name/	VPN 3000	ASA	PIX	Syntax/ Type	Single or Multi-Valued	Possible Values
L2TP-Encryption	Y			Integer	Single	Bitmap: 1 = Encryption required 2 = 40 bit 4 = 128 bits 8 = Stateless-Req 15 = 40/128-Encr/Stateless-Req
L2TP-MPPC-Compression	Y			Integer	Single	0 = Disabled 1 = Enabled
MS-Client-Subnet-Mask	Y	Y	Y	String	Single	An IP address
PFS-Required	Y	Y	Y	Boolean	Single	0 = No 1 = Yes
Port-Forwarding-Name	Y	Y		String	Single	Name string (for example, “Corporate-Apps”)
PPTP-Encryption	Y			Integer	Single	Bitmap: 1 = Encryption required 2 = 40 bits 4 = 128 bits 8 = Stateless-Required Example: 15 = 40/128-Encr/Stateless-Req
PPTP-MPPC-Compression	Y			Integer	Single	0 = Disabled 1 = Enabled
Primary-DNS	Y	Y	Y	String	Single	An IP address
Primary-WINS	Y	Y	Y	String	Single	An IP address
Privilege-Level						
Required-Client-Firewall-Vendor-Code	Y	Y	Y	Integer	Single	1 = Cisco Systems (with Cisco Integrated Client) 2 = Zone Labs 3 = NetworkICE 4 = Sygate 5 = Cisco Systems (with Cisco Intrusion Prevention Security Agent)
Required-Client-Firewall-Description	Y	Y	Y	String	Single	String

Table D-2 Security Appliance Supported Cisco Attributes for LDAP Authorization (continued)

Attribute Name/	VPN 3000	ASA	PIX	Syntax/ Type	Single or Multi-Valued	Possible Values
Required-Client-Firewall-Product-Code	Y	Y	Y	Integer	Single	Cisco Systems Products: 1 = Cisco Intrusion Prevention Security Agent or Cisco Integrated Client (CIC) Zone Labs Products: 1 = Zone Alarm 2 = Zone AlarmPro 3 = Zone Labs Integrity NetworkICE Product: 1 = BlackIce Defender/Agent Sygate Products: 1 = Personal Firewall 2 = Personal Firewall Pro 3 = Security Agent
Require-HW-Client-Auth	Y	Y	Y	Boolean	Single	0 = Disabled 1 = Enabled
Require-Individual-User-Auth	Y	Y	Y	Integer	Single	0 = Disabled 1 = Enabled
Secondary-DNS	Y	Y	Y	String	Single	An IP address
Secondary-WINS	Y	Y	Y	String	Single	An IP address
SEP-Card-Assignment				Integer	Single	Not used
Simultaneous-Logins	Y	Y	Y	Integer	Single	0-2147483647
Strip-Realm	Y	Y	Y	Boolean	Single	0 = Disabled 1 = Enabled
TACACS-Authtype	Y	Y	Y	Interger	Single	
TACACS-Privilege-Level	Y	Y	Y	Interger	Single	
Tunnel-Group-Lock		Y	Y	String	Single	Name of the tunnel group or “none”
Tunneling-Protocols	Y	Y	Y	Integer	Single	1 = PPTP 2 = L2TP 4 = IPSec 8 = L2TP/IPSec 16 = WebVPN. 8 and 4 are mutually exclusive (0 - 11, 16 - 27 are legal values)
Use-Client-Address	Y			Boolean	Single	0 = Disabled 1 = Enabled
User-Auth-Server-Name	Y			String	Single	IP address or hostname
User-Auth-Server-Port	Y			Integer	Single	Port number for server protocol
User-Auth-Server-Secret	Y			String	Single	Server password
WebVPN-ACL-Filters		Y		String	Single	Access-List name

Table D-2 Security Appliance Supported Cisco Attributes for LDAP Authorization (continued)

Attribute Name/	VPN 3000	ASA	PIX	Syntax/ Type	Single or Multi-Valued	Possible Values
WebVPN-Apply-ACL-Enable	Y	Y		Integer	Single	0 = Disabled 1 = Enabled
WebVPN-Citrix-Support-Enable	Y	Y		Integer	Single	0 = Disabled 1 = Enabled
WebVPN-Content-Filter-Parameters	Y	Y		Integer	Single	1 = Java & ActiveX 2 = Java scripts 4 = Images 8 = Cookies in images Add the values to filter multiple parameters. For example: enter 10 to filter both Java scripts and cookies. (10 = 2 + 8)
WebVPN-Enable-functions				Integer	Single	Not used - deprecated
WebVPN-Exchange-Server-Address				String	Single	Not used - deprecated
WebVPN-Exchange-Server-NETBIOS-Name				String	Single	Not used - deprecated
WebVPN-File-Access-Enable	Y	Y		Integer	Single	0 = Disabled 1 = Enabled
WebVPN-File-Server-Browsing-Enable	Y	Y		Integer	Single	0 = Disabled 1 = Enabled
WebVPN-File-Server-Entry-Enable	Y	Y		Integer	Single	0 = Disabled 1 = Enabled
WebVPN-Forwarded-Ports		Y		String	Single	Port-Forward list name
WebVPN-Homepage	Y	Y		String	Single	A URL such as http://example-portal.com.
WebVPN-Macro-Substitution-Value1	Y	Y		String	Single	
WebVPN-Macro-Substitution-Value2	Y	Y		String	Single	
WebVPN-Port-Forwarding-Auto-Download-Enable	Y	Y		Integer	Single	0 = Disabled 1 = Enabled
WebVPN-Port-Forwarding- Enable	Y	Y		Integer	Single	0 = Disabled 1 = Enabled
WebVPN-Port-Forwarding-Exchange-Proxy-Enable	Y	Y		Integer	Single	0 = Disabled 1 = Enabled
WebVPN-Port-Forwarding-HTTP-Proxy-Enable	Y	Y		Integer	Single	0 = Disabled 1 = Enabled
WebVPN-Single-Sign-On-Server-Name		Y		String	Single	Name of the SSO Server (1 - 31 characters).

Table D-2 Security Appliance Supported Cisco Attributes for LDAP Authorization (continued)

Attribute Name/	VPN 3000	ASA	PIX	Syntax/ Type	Single or Multi-Valued	Possible Values
WebVPN-SVC-Client-DPD	Y	Y		Integer	Single	0 = Disabled n = Dead Peer Detection value in seconds (30 - 3600)
WebVPN-SVC-Compression	Y	Y		Integer	Single	0 = None 1 = Deflate Compression
WebVPN-SVC-Enable	Y	Y		Integer	Single	0 = Disabled 1 = Enabled
WebVPN-SVC-Gateway-DPD	Y	Y		Integer	Single	0 = Disabled n = Dead Peer Detection value in seconds (30 - 3600)
WebVPN-SVC-Keepalive	Y	Y		Integer	Single	0 = Disabled n = Keepalive value in seconds (15 - 600)
WebVPN-SVC-Keep-Enable	Y	Y		Integer	Single	0 = Disabled 1 = Enabled
WebVPN-SVC-Rekey-Method	Y	Y		Integer	Single	0 = None 1 = SSL 2 = New tunnel 3 = Any (sets to SSL)
WebVPN-SVC-Rekey-Period	Y	Y		Integer	Single	0 = Disabled n = Retry period in minutes (4 - 10080)
WebVPN-SVC-Required-Enable	Y	Y		Integer	Single	0 = Disabled 1 = Enabled
WebVPN-URL-Entry-Enable	Y	Y		Integer	Single	0 = Disabled 1 = Enabled
WebVPN-URL-List		Y		String	Single	URL-list name

Cisco-AV-Pair Attribute Syntax

The syntax of each Cisco-AV-Pair rule is as follows:

[Prefix] [Action] [Protocol] [Source] [Source Wildcard Mask] [Destination] [Destination Wildcard Mask] [Established] [Log] [Operator] [Port]

Table D-3 describes the syntax rules.

Table D-3 AV-Pair Attribute Syntax Rules

Field	Description
Prefix	A unique identifier for the AV pair. For example: `ip:inacl#1=` (for standard access lists) or `webvpn:inacl#` (for clientless SSL VPN access lists). This field only appears when the filter has been sent as an AV pair.
Action	Action to perform if rule matches: deny, permit.

Table D-3 AV-Pair Attribute Syntax Rules

Field	Description
Protocol	Number or name of an IP protocol. Either an integer in the range 0 - 255 or one of the following keywords: icmp, igmp, ip, tcp, udp.
Source	Network or host that sends the packet. Specify it as an IP address, a hostname, or the keyword "any." If using an IP address, the source wildcard mask must follow.
Source Wildcard Mask	The wildcard mask that applies to the source address.
Destination	Network or host that receives the packet. Specify as an IP address, a hostname, or the keyword "any." If using an IP address, the source wildcard mask must follow.
Destination Wildcard Mask	The wildcard mask that applies to the destination address.
Log	Generates a FILTER log message. You must use this keyword to generate events of severity level 9.
Operator	Logic operators: greater than, less than, equal to, not equal to.
Port	The number of a TCP or UDP port in the range 0 - 65535.

For example:

```
ip:inacl#1=deny ip 10.155.10.0 0.0.0.255 10.159.2.0 0.0.0.255 log
ip:inacl#2=permit TCP any host 10.160.0.1 eq 80 log

webvpn:inacl#1=permit url http://www.website.com
webvpn:inacl#2=deny smtp any host 10.1.3.5
webvpn:inacl#3=permit url cifs://mar_server/peopleshare1
```

Note Use Cisco-AV pair entries with the ip:inacl# prefix to enforce access lists for remote IPSec and SSL VPN Client (SVC) tunnels.

Use Cisco-AV pair entries with the webvpn:inacl# prefix to enforce access lists for SSL VPN clientless (browser-mode) tunnels.

Table D-4 lists the tokens for the Cisco-AV-pair attribute:

Table D-4 Security Appliance-Supported Tokens

Token	Syntax Field	Description
ip:inacl#*Num*=	N/A (Identifier)	(Where *Num* is a unique integer.) Starts all AV pair access control lists. Enforces access lists for remote IPSec and SSL VPN (SVC) tunnels.
webvpn:inacl#*Num*=	N/A (Identifier)	(Where *Num* is a unique integer.) Starts all clientless SSL AV pair access control lists. Enforces access lists for clientless (browser-mode) tunnels.
deny	Action	Denies action. (Default)
permit	Action	Allows action.
icmp	Protocol	Internet Control Message Protocol (ICMP)
1	Protocol	Internet Control Message Protocol (ICMP)

Table D-4 Security Appliance-Supported Tokens

Token	Syntax Field	Description
IP	Protocol	Internet Protocol (IP)
0	Protocol	Internet Protocol (IP)
TCP	Protocol	Transmission Control Protocol (TCP)
6	Protocol	Transmission Control Protocol (TCP)
UDP	Protocol	User Datagram Protocol (UDP)
17	Protocol	User Datagram Protocol (UDP)
any	Hostname	Rule applies to any host.
host	Hostname	Any alpha-numeric string that denotes a hostname.
log	Log	When the event is hit, a filter log message appears. (Same as permit and log or deny and log.)
lt	Operator	Less than value
gt	Operator	Greater than value
eq	Operator	Equal to value
neq	Operator	Not equal to value
range	Operator	Inclusive range. Should be followed by two values.

Active Directory/LDAP VPN Remote Access Authorization Use Cases

This section presents example procedures for configuring authentication and authorization on the security appliance using the Microsoft Active Directory server. It includes the following use cases:

- User-Based Attributes Policy Enforcement, page D-15
- Placing LDAP users in a specific Group-Policy, page D-17
- Enforcing Static IP Address Assignment for AnyConnect Tunnels, page D-19
- Enforcing Dial-in Allow or Deny Access, page D-22
- Enforcing Logon Hours and Time-of-Day Rules, page D-25

Other configuration examples available on Cisco.com include the following TechNotes:

- *ASA/PIX: Mapping VPN Clients to VPN Group Policies Through LDAP Configuration Example* at:

  http://www.cisco.com/en/US/products/ps6120/products_configuration_example09186a008089149d.shtml

- *PIX/ASA 8.0: Use LDAP Authentication to Assign a Group Policy at Login* at:

  http://www.cisco.com/en/US/partner/products/ps6120/products_configuration_example09186a00808d1a7c.shtml

User-Based Attributes Policy Enforcement

Any standard LDAP attribute can be mapped to a well-known Vendor Specific Attribute (VSA) Likewise, one or more LDAP attribute(s) can be mapped to one or more Cisco LDAP attributes.

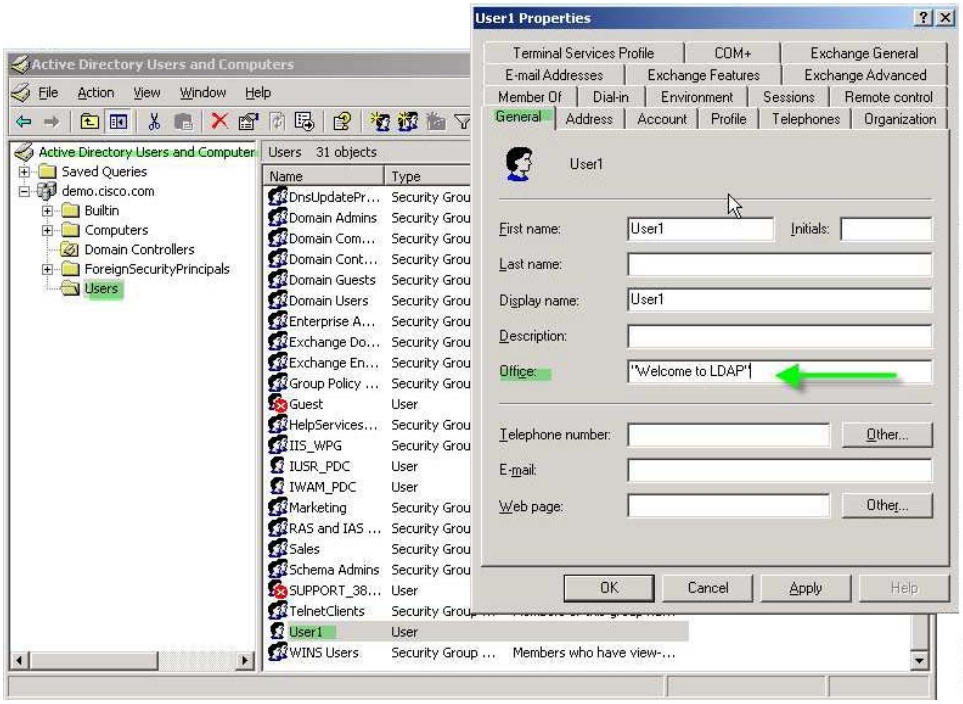
In this use case we configure the security appliance to enforce a simple banner for a user configured on an AD LDAP server. For this case, on the server, we use the Office field in the General tab to enter the banner text. This field uses the attribute named *physicalDeliveryOfficeName*. On the security appliance, we create an attribute map that maps *physicalDeliveryOfficeName* to the Cisco attribute *Banner1*. During authentication, the security appliance retrieves the value of physicalDeliveryOfficeName from the server, maps the value to the Cisco attribute Banner1, and displays the banner to the user.

This case applies to any connection type, including the IPSec VPN client, AnyConnect SSL VPN client, or clientless SSL VPN. For the purposes of this case, User1 is connecting through a clientless SSL VPN connection.

Step 1 Configure the attributes for a user on the AD/LDAP Server.

Right-click a user. The properties window displays (Figure D-3). Click the General tab and enter some banner text in the Office field. The Office field uses the AD/LDAP attribute *physicalDeliveryOfficeName*.

Figure D-3 Figure 3 LDAP User configuration



Step 2 Create an LDAP attribute map on the security appliance:

The following example creates the map *Banner*, and maps the AD/LDAP attribute *physicalDeliveryOfficeName* to the Cisco attribute *Banner1*:

```
hostname(config)# ldap attribute-map Banner
hostname(config-ldap-attribute-map)# map-name physicalDeliveryOfficeName Banner1
```

Step 3 Associate the LDAP attribute map to the AAA server.

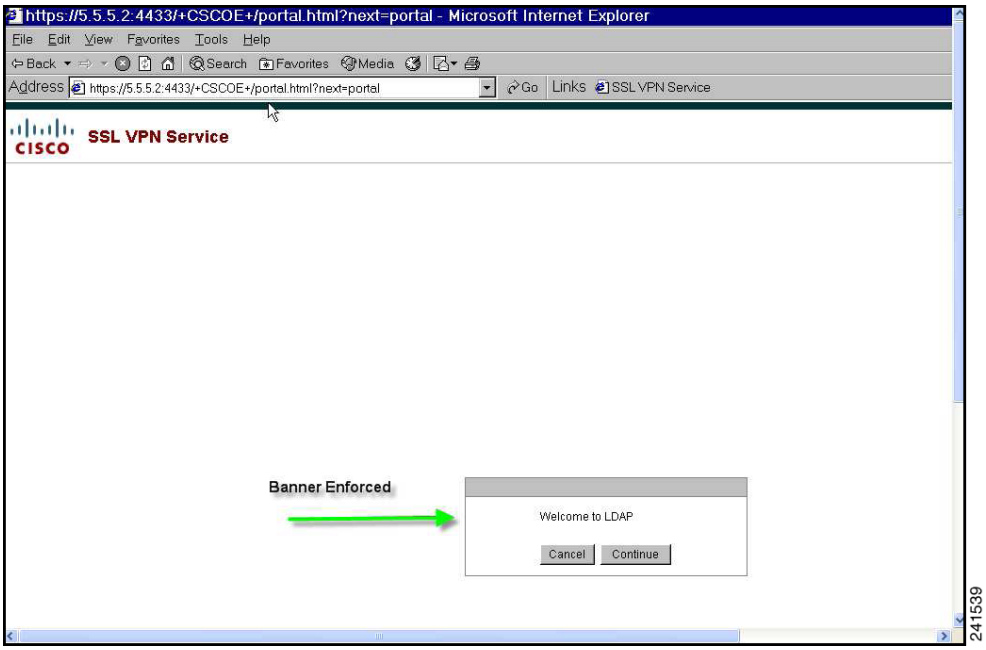
The following example enters the aaa server host configuration more for the host *3.3.3.4*, in the AAA server group *MS_LDAP*, and associates the attribute map *Banner* that you created in step 2:

```
hostname(config)# aaa-server MS_LDAP host 3.3.3.4
hostname(config-aaa-server-host)# ldap-attribute-map Banner
```

Step 4 Test the banner enforcement.

This example shows a clientless SSL connection and the banner enforced through the attribute map after the user authenticates (Figure D-4).

Figure D-4 Banner Displayed



Placing LDAP users in a specific Group-Policy

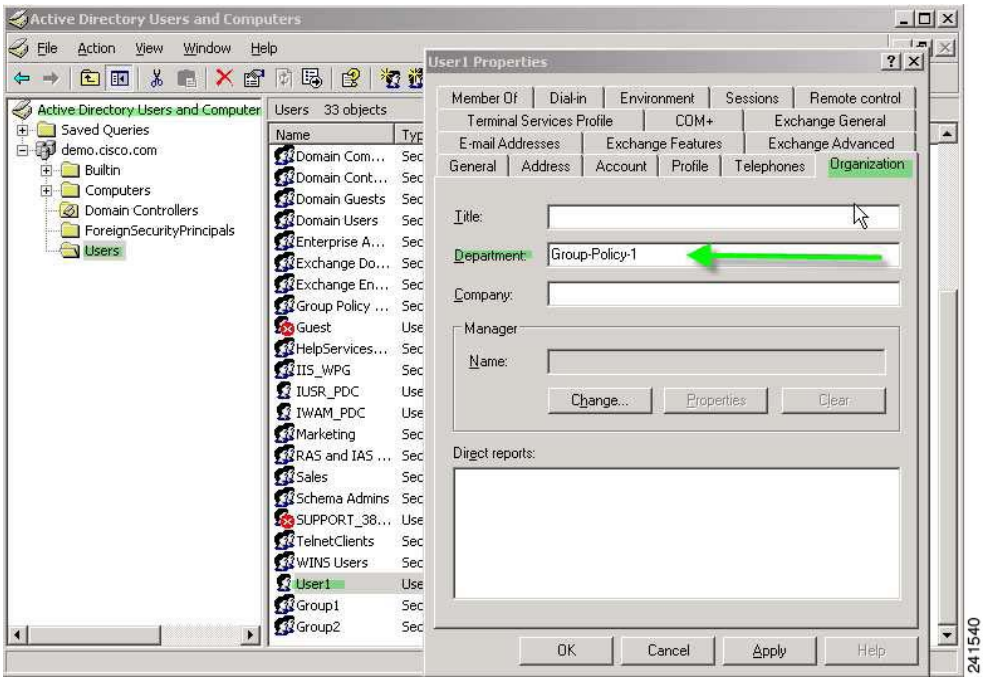
In this case we authenticate User1 on the AD LDAP server to a specific group policy on the security appliance. On the server, we use the *Department* field of the Organization tab to enter the name of the group policy. Then we create an attribute map and map Department to the Cisco attribute *IETF-Radius-Class*. During authentication, the security appliance retrieves the value of Department from the server, maps the value to the IETF-Radius-Class, and places User1 in the group policy.

This case applies to any connection type, including the IPSec VPN client, AnyConnect SSL VPN client, or clientless SSL VPN. For the purposes of this case, user1 is connecting through a clientless SSL VPN connection.

Step 1 Configure the attributes for the user on the AD LDAP Server.

Right-click the user. The Properties window displays (Figure D-5). Click the Organization tab and enter *Group-Policy-1* in the Department field.

Figure D-5 AD LDAP Department attribute



Step 2 Define an attribute map for the LDAP configuration shown in Step 1.

In this case we map the AD attribute Department to the Cisco attribute IETF-Radius-Class. For example:

```
hostname(config)# ldap attribute-map group_policy
hostname(config-ldap-attribute-map)# map-name Department IETF-Radius-Class
```

Step 3 Associate the LDAP attribute map to the AAA server.

The following example enters the aaa server host configuration mode for the host *3.3.3.4*, in the AAA server group *MS_LDAP*, and associates the attribute map *group_policy* that you created in step 2:

```
hostname(config)# aaa-server MS_LDAP host 3.3.3.4
hostname(config-aaa-server-host)# ldap-attribute-map group_policy
```

Step 4 Add the new group-policy on the security appliance and configure the required policy attributes that will be assigned to the user. For this case, we created the Group-policy-1, the name entered in the Department field on the server:

```
hostname(config)# group-policy Group-policy-1 external server-group LDAP_demo
hostname(config-aaa-server-group)#
```

Step 5 Establish the VPN connection as the user would, and verify that the session inherits the attributes from Group-Policy1 (and any other applicable attributes from the default group-policy)

You can monitor the communication between the security appliance and the server by enabling the **debug ldap 255** command from privileged EXEC mode. Below is sample output of this command. The output has been edited to provide the key messages:

[29] Authentication successful for user1 to 3.3.3.4

[29] Retrieving user attributes from server 3.3.3.4

[29] Retrieved Attributes:

[29] department: value = Group-Policy-1

[29] mapped to IETF-Radius-Class: value = Group-Policy-1

Enforcing Static IP Address Assignment for AnyConnect Tunnels

In this case we configure the AnyConnect client user *Web1* to receive a static IP Address. We enter the address in the *Assign Static IP Address* field of the Dialin tab on the AD LDAP server. This field uses the *msRADIUSFramedIPAddress* attribute. We create an attribute map that maps it to the Cisco attribute *IETF-Radius-Framed-IP-Address*.

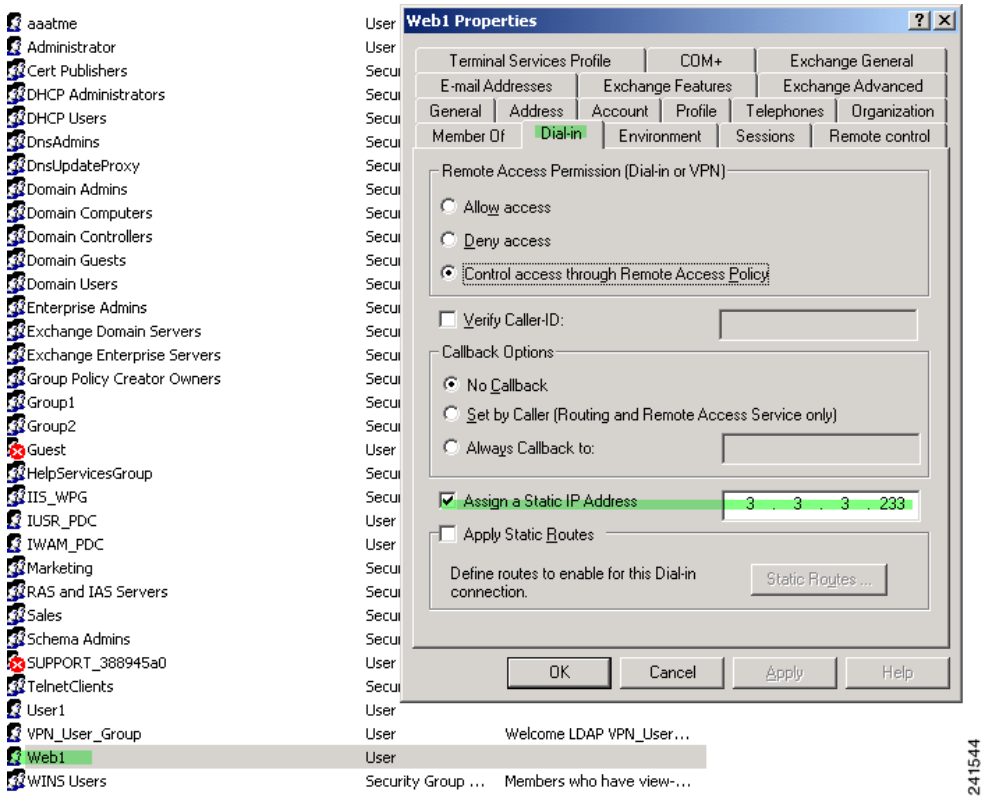
During authentication, the security appliance retrieves the value of msRADIUSFramedIPAddress from the server, maps the value to the Cisco attribute IETF-Radius-Framed-IP-Address, and provides the static address to User1 .

This case applies to full-tunnel clients, including the IPSec client and the SSL VPN clients (AnyConnect client 2.x and the legacy SSL VPN client).

Step 1 Configure the user attributes on the AD LDAP server.

Right-click on the user name. The Properties window displays (Figure D-6). Click the Dialin tab, check *Assign Static IP Address*, and enter an IP address. For this case we use 3.3.3.233.

Figure D-6 Assign Static IP Address



Step 2 Create an attribute map for the LDAP configuration shown in Step 1.

In this case we map the AD attribute *msRADIUSFrameIPAddress* used by the Static Address field to the Cisco attribute *IETF-Radius-Framed-IP-Address*.

For example:

```
hostname(config)# ldap attribute-map static_address
hostname(config-ldap-attribute-map)# map-name msRADIUSFrameIPAddress
IETF-Radius-Framed-IP-Address
```

Step 3 Associate the LDAP attribute map to the AAA server.

The following example enters the aaa server host configuration mode for the host *3.3.3.4*, in the AAA server group *MS_LDAP*, and associates the attribute map *static_address* that you created in step 2:

```
hostname(config)# aaa-server MS_LDAP host 3.3.3.4
hostname(config-aaa-server-host)# ldap-attribute-map static_address
```

Step 4 Verify the **vpn-address-assigment** command is configured to specify aaa by viewing this part of the configuration with the **show run all vpn-addr-assign command**:

vpn-addr-assign aaa

```
hostname(config)# show run all vpn-addr-assign
vpn-addr-assign aaa   <<<< ensure this configured.
no vpn-addr-assign dhcp
vpn-addr-assign local
hostname(config)#
```

Step 5 Establish a connection to the security appliance with the AnyConnect client. Observe the following:

- The banner is received in the same sequence as a clientless connection (Figure D-7).
- The user receives the IP address configured on the server and mapped to the security appliance (Figure D-8).

Figure D-7 Verify the Banner for the AnyConnect Session

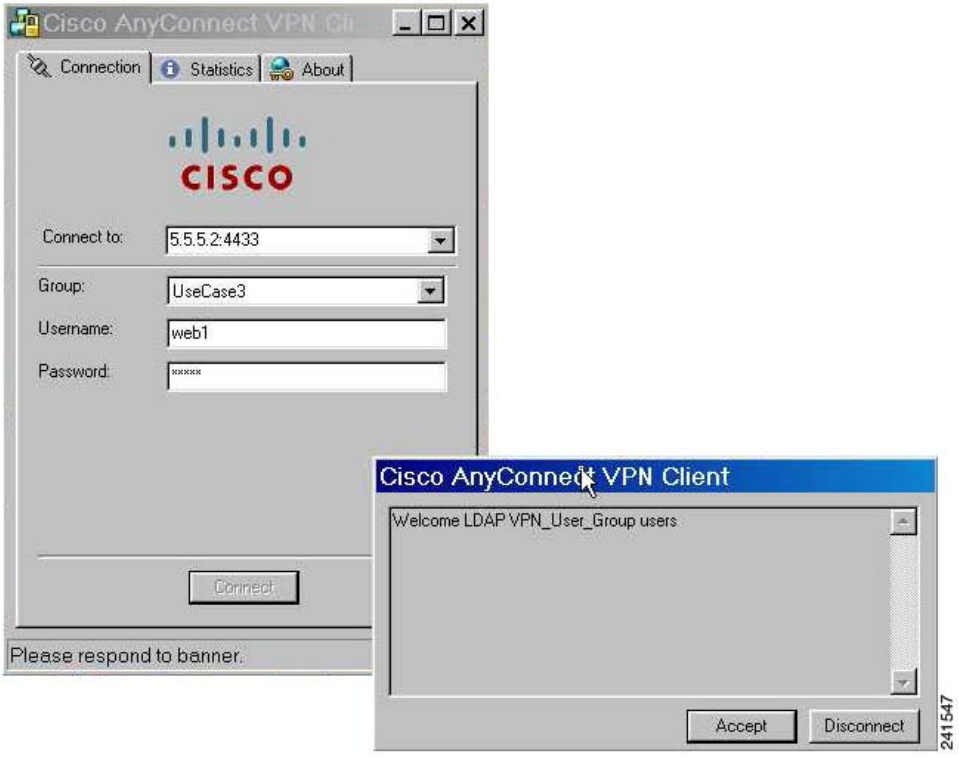
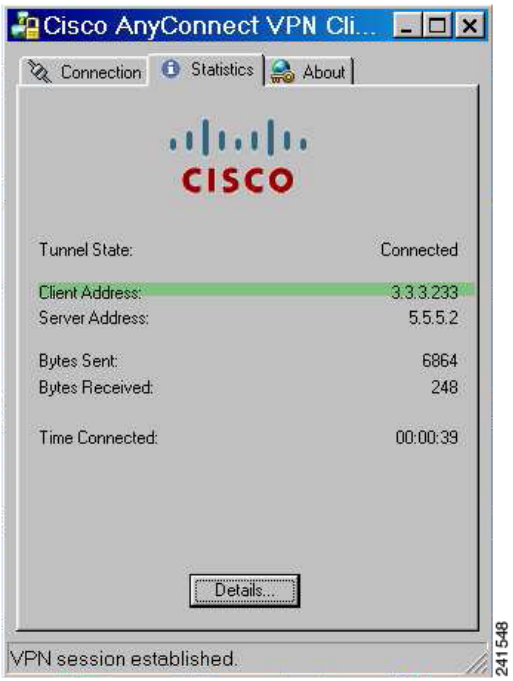


Figure D-8 AnyConnect Session Established



You can use the **show vpn-sessiondb svc** command to view the session details and verify the address assigned. You can also view the elapsed time since the AnyConnect client lost connectivity. If the session is active, 00:00m:00s appears in this Inactivity field.

```
hostname# show vpn-sessiondb svc

Session Type: SVC
Username     : web1                   Index        : 31
Assigned IP  : 3.3.3.233              Public IP    : 10.86.181.70
Protocol     : Clientless SSL-Tunnel DTLS-Tunnel
Encryption   : RC4 AES128             Hashing      : SHA1
Bytes Tx     : 304140                 Bytes Rx     : 470506
Group Policy : VPN_User_Group         Tunnel Group : UseCase3_TunnelGroup
Login Time   : 11:13:05 UTC Tue Aug 28 2007
Duration     : 0h:01m:48s
Inactivity   :  0h:28m:48s
NAC Result   : Unknown
VLAN Mapping : N/A                    VLAN         : none

BXB-ASA5540#
```

Enforcing Dial-in Allow or Deny Access

In this case, we create an LDAP attribute map that specifies the tunneling protocols allowed by the user. We map the Allow Access and Deny Access settings on the Dialin tab to the Cisco attribute Tunneling-Protocols. The Cisco Tunneling-Protocols supports the bit-map values shown in Table D-5:

Table D-5 Bitmap Values for Cisco Tunneling-Protocol Attribute

Value	Tunneling Protocol
1	PPTP
2	L2TP
4[1]	IPSec
8[2]	L2TP/IPSEC
16	clientless SSL
32	SSL Client—AnyConnect or legacy SSL VPN client

1. IPSec and L2TP over IPSec are not supported simultaneously. Therefore, the values 4 and 8 are mutually exclusive.
2. See note 1.

Using this attribute, we create an Allow Access (TRUE) or a Deny Access (FALSE) condition for the protocols and enforce what method the user is allowed access with.

For this simplified example, by mapping the tunnel-protocol IPSec (4), we can create an allow (true) condition for the IPSec Client. We also map WebVPN (16) and SVC/AC (32) which is mapped as value of 48 (16+32) and create a deny (false) condition. This allows the user to connect to the security appliance using IPSec, but any attempt to connect using clientless SSL or the AnyConnect client is denied.

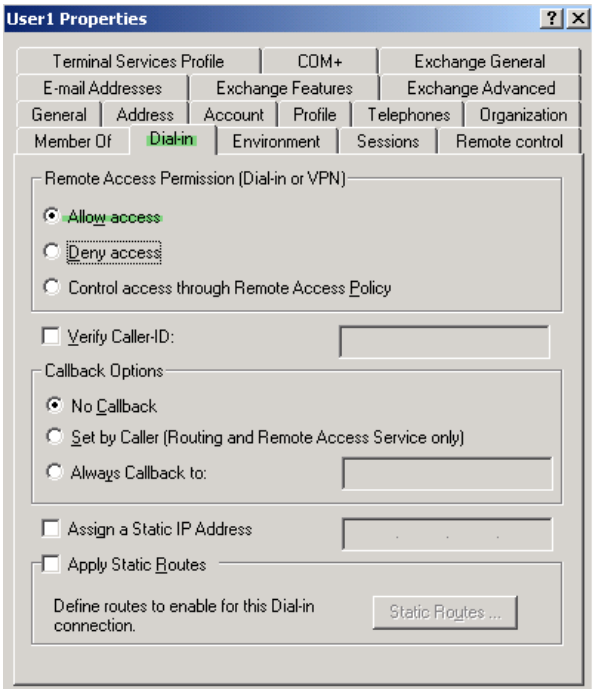
Another example of enforcing Dial-in Allow Acess or Deny Access can be found in the Tech Note *ASA/PIX: Mapping VPN Clients to VPN Group Policies Through LDAP Configuration Example,* at this URL:

http://www.cisco.com/en/US/products/ps6120/products_configuration_example09186a008089149d.shtml

Step 1 Configure the user attributes on the AD LDAP server.

Right-click on the user. The Properties window displays. Click the Dial-in tab. Select **Allow Access** (Figure D-9).

Figure D-9 AD-LDAP user1 - Allow access



Note If you select the third option "Control access through the Remote Access Policy", then a value is not returned from the server, and the permissions that are enforced are based on the internal group policy settings of the security appliance.

Step 2 Create an attribute map to allow both an IPSec and AnyConnect connection, but deny a clientless SSL connection.

In this case we create the map *tunneling_protocols*, and map the AD attribute *msNPAllowDialin* used by the Allow Access setting to the Cisco attribute *Tunneling-Protocols* using the **map-name** command, and add map values with the **map-value** command,

For example:

```
hostname(config)# ldap attribute-map tunneling_protocols
hostname(config-ldap-attribute-map)# map-name msNPAllowDialin Tunneling-Protocols
hostname(config-ldap-attribute-map)# map-value msNPAllowDialin FALSE  48
hostname(config-ldap-attribute-map)# map-value msNPAllowDialin TRUE 4
```

Step 3 Associate the LDAP attribute map to the AAA server.

The following example enters the aaa server host configuration mode for the host *3.3.3.4*, in the AAA server group *MS_LDAP*, and associates the attribute map *tunneling_protocols* that you created in step 2:

```
hostname(config)# aaa-server MS_LDAP host 3.3.3.4
hostname(config-aaa-server-host)# ldap-attribute-map tunneling_protocols
```

Step 4 Verify the attribute map works as configured.

Using a PC as a remote user would, attempt connections using clientless SSL, the AnyConnect client, and the IPSec client. The clientless and AnyConnect connections should fail and the user should be informed that an unauthorized connection mechanism was the reason for the failed connection. The IPSec client should connect because IPSec is an allowed tunneling protocol according to attribute map.

Figure D-10 Login Denied Message for Clientless User

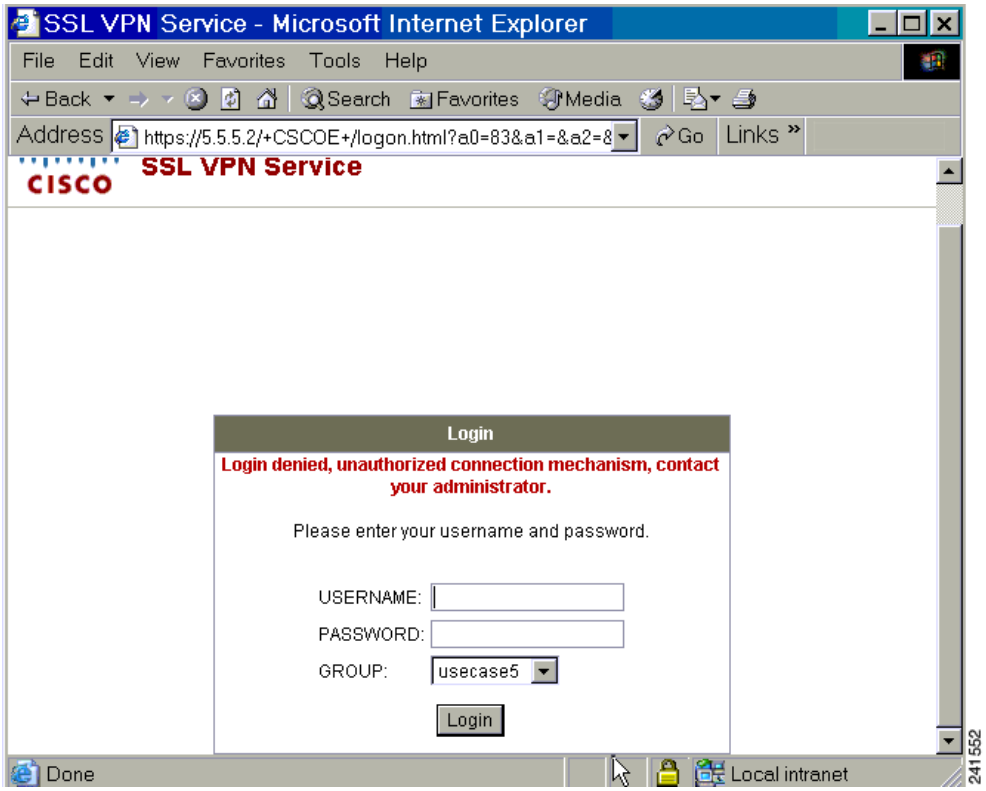
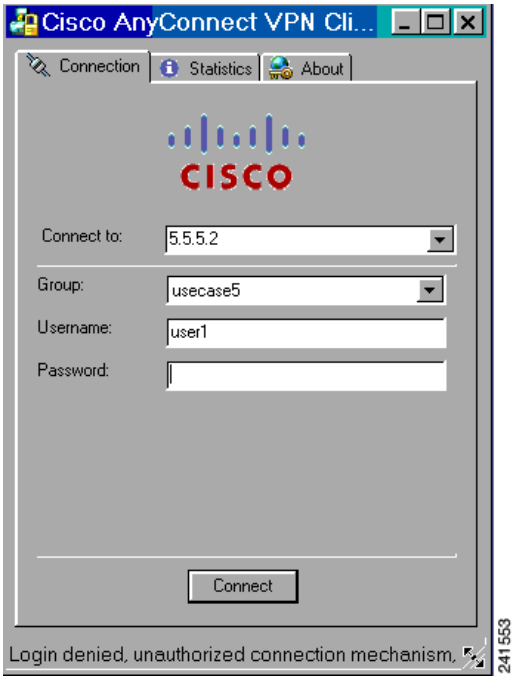


Figure D-11 Login Denied Message for AnyConnect Client User.



Enforcing Logon Hours and Time-of-Day Rules

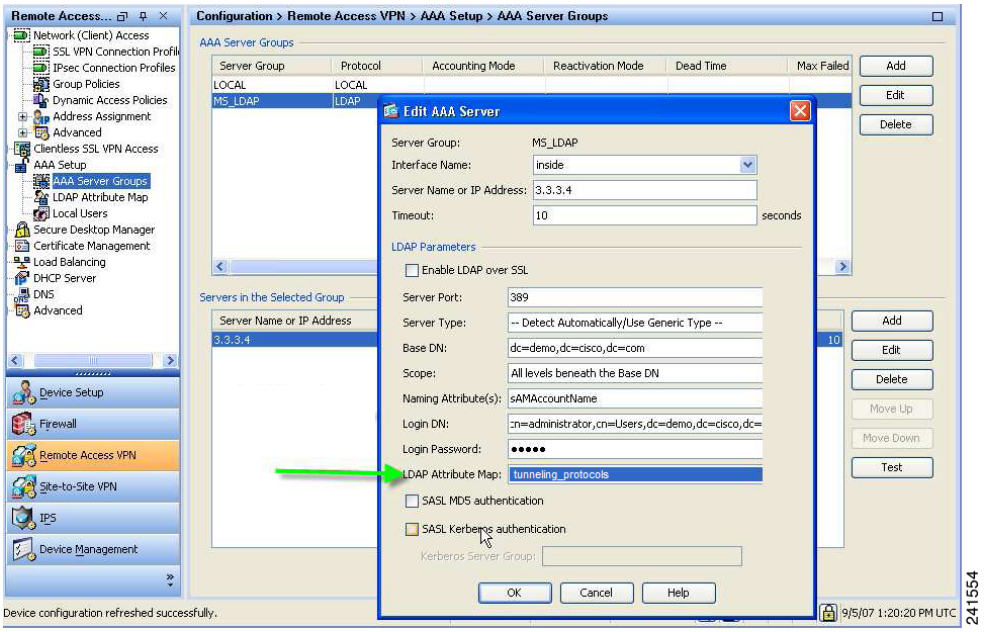
In this use case we configure and enforce the hours that a clientless SSL user is allowed to access the network. A good example of this is when you want to allow a business partner access to the network only during normal business hours.

For this case, on the AD server, we use the *Office* field to enter the name of the partner. This field uses the *physicalDeliveryOfficeName* attribute. Then we create an attribute map on the security appliance to map that attribute to the Cisco attribute *Access-Hours*. During authentication, the security appliance retrieves the value of physicalDeliveryOfficeName *(*the Office field*)* and maps it to Access-Hours.

Step 1 Configure the user attributes on the AD LDAP server.

Select the user. Right click on Properties. The Properties window displays (Figure D-12). For this case, we use the Office field of the General tab:

Figure D-12 Active Directory - Time-range



Step 2 Create an attribute map.

In this case we create the attribute map access_hours and map the AD attribute *physicalDeliveryOfficeName* used by the Office field to the Cisco attribute *Access-Hours*.

For example:

```
hostname(config)# ldap attribute-map access_hours
hostname(config-ldap-attribute-map)# map-name physicalDeliveryOfficeName Access-Hours
```

Step 3 Associate the LDAP attribute map to the AAA server.

The following example enters the aaa server host configuration mode for the host *3.3.3.4*, in the AAA server group *MS_LDAP*, and associates the attribute map *access_hours* that you created in step 2:

```
hostname(config)# aaa-server MS_LDAP host 3.3.3.4
hostname(config-aaa-server-host)# ldap-attribute-map access_hours
```

Step 4 Configure time ranges for each value allowed on the server. In this case, we entered Partner in the Office field for User1. Therefore, there must be a time range configured for Partner. The following example configures Partner access hours from 9am to 5pm Monday through Friday:

```
hostname(config)# time-range Partner
hostname(config-time-range)# periodic weekdays 09:00 to 17:00
```

Configuring an External RADIUS Server

This section presents an overview of the RADIUS configuration procedure and defines the Cisco RADIUS attributes. It includes the following topics:

- Reviewing the RADIUS Configuration Procedure, page D-27
- Security Appliance RADIUS Authorization Attributes, page D-27

Reviewing the RADIUS Configuration Procedure

This section describes the RADIUS configuration steps required to support authentication and authorization of the security appliance users. Follow these steps to set up the RADIUS server to inter operate with the security appliance.

Step 1 Load the security appliance attributes into the RADIUS server. The method you use to load the attributes depends on which type of RADIUS server you are using:

- If you are using Cisco ACS: the server already has these attributes integrated. You can skip this step.
- If you are using a FUNK RADIUS server: Cisco supplies a dictionary file that contains all the security appliance attributes. Obtain this dictionary file, `cisco3k.dct`, from Software Center on CCO or from the security appliance CD-ROM. Load the dictionary file on your server.
- For other vendors' RADIUS servers (for example, Microsoft Internet Authentication Service): you must manually define each security appliance attribute. To define an attribute, use the attribute name or number, type, value, and vendor code (3076). For a list of security appliance RADIUS authorization attributes and values, see Table D-6.

Step 2 Set up the users or groups with the permissions and attributes to send during IPSec or SSL tunnel establishment.

Security Appliance RADIUS Authorization Attributes

Authorization refers to the process of enforcing permissions or attributes. A RADIUS server defined as an authentication server enforces permissions or attributes if they are configured.

Table D-6 lists all the possible security appliance supported RADIUS attributes that can be used for user authorization.

Note RADIUS attribute names do not contain the cVPN3000 prefix. Cisco Secure ACS 4.x supports this new nomenclature, but attribute names in pre-4.0 ACS releases still include the cVPN3000 prefix. The appliances enforce the RADIUS attributes based on attribute numeric ID, not attribute name. LDAP attributes are enforced by their name, not by the ID.

Table D-6 Security Appliance Supported RADIUS Attributes and Values

Attribute Name	VPN 3000	ASA	PIX	Attr. #	Syntax/ Type	Single or Multi-Valued	Description or Value
Access-Hours	Y	Y	Y	1	String	Single	Name of the time range, for example, Business-hours
Simultaneous-Logins	Y	Y	Y	2	Integer	Single	An integer 0 to 2147483647
Primary-DNS	Y	Y	Y	5	String	Single	An IP address
Secondary-DNS	Y	Y	Y	6	String	Single	An IP address
Primary-WINS	Y	Y	Y	7	String	Single	An IP address
Secondary-WINS	Y	Y	Y	8	String	Single	An IP address
SEP-Card-Assignment				9	Integer	Single	Not used
Tunneling-Protocols	Y	Y	Y	11	Integer	Single	1 = PPTP 2 = L2TP 4 = IPSec 8 = L2TP/IPSec 16 = WebVPN 4 and 8 are mutually exclusive; 0-11 and 16-27 are legal values.
IPSec-Sec-Association	Y			12	String	Single	Name of the security association
IPSec-Authentication	Y			13	Integer	Single	0 = None 1 = RADIUS 2 = LDAP (authorization only) 3 = NT Domain 4 = SDI 5 = Internal 6 = RADIUS with Expiry 7 = Kerberos/Active Directory
Banner1	Y	Y	Y	15	String	Single	Banner string
IPSec-Allow-Passwd-Store	Y	Y	Y	16	Boolean	Single	0 = Disabled 1 = Enabled
Use-Client-Address	Y			17	Boolean	Single	0 = Disabled 1 = Enabled

Table D-6 Security Appliance Supported RADIUS Attributes and Values (continued)

Attribute Name	VPN 3000	ASA	PIX	Attr. #	Syntax/ Type	Single or Multi-Valued	Description or Value
PPTP-Encryption	Y			20	Integer	Single	Bitmap: 1 = Encryption required 2 = 40 bits 4 = 128 bits 8 = Stateless-Required 15 = 40/128-Encr/Stateless-Req
L2TP-Encryption	Y			21	Integer	Single	Bitmap: 1 = Encryption required 2 = 40 bit 4 = 128 bits 8 = Stateless-Req 15 = 40/128-Encr/Stateless-Req
IPSec-Split-Tunnel-List	Y	Y	Y	27	String	Single	Specifies the name of the network/access list that describes the split tunnel inclusion list
IPSec-Default-Domain	Y	Y	Y	28	String	Single	Specifies the single default domain name to send to the client (1-255 characters)
IPSec-Split-DNS-Names	Y	Y	Y	29	String	Single	Specifies the list of secondary domain names to send to the client (1-255 characters)
IPSec-Tunnel-Type	Y	Y	Y	30	Integer	Single	1 = LAN-to-LAN 2 = Remote access
IPSec-Mode-Config	Y	Y	Y	31	Boolean	Single	0 = Disabled 1 = Enabled
IPSec-User-Group-Lock	Y			33	Boolean	Single	0 = Disabled 1 = Enabled
IPSec-Over-UDP	Y	Y	Y	34	Boolean	Single	0 = Disabled 1 = Enabled
IPSec-Over-UDP-Port	Y	Y	Y	35	Integer	Single	4001 - 49151, default = 10000
Banner2	Y	Y	Y	36	String	Single	A banner string that is concatenated to the Banner1 string, if configured.

Table D-6 Security Appliance Supported RADIUS Attributes and Values (continued)

Attribute Name	VPN 3000	ASA	PIX	Attr. #	Syntax/ Type	Single or Multi-Valued	Description or Value
PPTP-MPPC-Compression	Y			37	Integer	Single	0 = Disabled 1 = Enabled
L2TP-MPPC-Compression	Y			38	Integer	Single	0 = Disabled 1 = Enabled
IPSec-IP-Compression	Y	Y	Y	39	Integer	Single	0 = Disabled 1 = Enabled
IPSec-IKE-Peer-ID-Check	Y	Y	Y	40	Integer	Single	1 = Required 2 = If supported by peer certificate 3 = Do not check
IKE-Keep-Alives	Y	Y	Y	41	Boolean	Single	0 = Disabled 1 = Enabled
IPSec-Auth-On-Rekey	Y	Y	Y	42	Boolean	Single	0 = Disabled 1 = Enabled
Required-Client- Firewall-Vendor-Code	Y	Y	Y	45	Integer	Single	1 = Cisco Systems (with Cisco Integrated Client) 2 = Zone Labs 3 = NetworkICE 4 = Sygate 5 = Cisco Systems (with Cisco Intrusion Prevention Security Agent)

Table D-6 Security Appliance Supported RADIUS Attributes and Values (continued)

Attribute Name	VPN 3000	ASA	PIX	Attr. #	Syntax/ Type	Single or Multi-Valued	Description or Value
Required-Client-Firewall-Product-Code	Y	Y	Y	46	Integer	Single	Cisco Systems Products: 1 = Cisco Intrusion Prevention Security Agent or Cisco Integrated Client (CIC) Zone Labs Products: 1 = Zone Alarm 2 = Zone AlarmPro 3 = Zone Labs Integrity NetworkICE Product: 1 = BlackIce Defender/Agent Sygate Products: 1 = Personal Firewall 2 = Personal Firewall Pro 3 = Security Agent
Required-Client-Firewall-Description	Y	Y	Y	47	String	Single	String
Require-HW-Client-Auth	Y	Y	Y	48	Boolean	Single	0 = Disabled 1 = Enabled
Required-Individual-User-Auth	Y	Y	Y	49	Integer	Single	0 = Disabled 1 = Enabled
Authenticated-User-Idle-Timeout	Y	Y	Y	50	Integer	Single	1-35791394 minutes
Cisco-IP-Phone-Bypass	Y	Y	Y	51	Integer	Single	0 = Disabled 1 = Enabled
IPSec-Split-Tunneling-Policy	Y	Y	Y	55	Integer	Single	0 = No split tunneling 1 = Split tunneling 2 = Local LAN permitted
IPSec-Required-Client-Firewall-Capability	Y	Y	Y	56	Integer	Single	0 = None 1 = Policy defined by remote FW Are-You-There (AYT) 2 = Policy pushed CPP 4 = Policy from server
IPSec-Client-Firewall-Filter-Name	Y			57	String	Single	Specifies the name of the filter to be pushed to the client as firewall policy
IPSec-Client-Firewall-Filter-Optional	Y	Y	Y	58	Integer	Single	0 = Required 1 = Optional

Table D-6 Security Appliance Supported RADIUS Attributes and Values (continued)

Attribute Name	VPN 3000	ASA	PIX	Attr. #	Syntax/ Type	Single or Multi-Valued	Description or Value
IPSec-Backup-Servers	Y	Y	Y	59	String	Single	1 = Use Client-Configured list 2 = Disable and clear client list 3 = Use Backup Server list
IPSec-Backup-Server-List	Y	Y	Y	60	String	Single	Server Addresses (space delimited)
DHCP-Network-Scope	Y	Y	Y	61	String	Single	IP Address
Intercept-DHCP-Configure-Msg	Y	Y	Y	62	Boolean	Single	0 = Disabled 1 = Enabled
MS-Client-Subnet-Mask	Y	Y	Y	63	Boolean	Single	An IP address
Allow-Network-Extension-Mode	Y	Y	Y	64	Boolean	Single	0 = Disabled 1 = Enabled
Authorization-Type	Y	Y	Y	65	Integer	Single	0 = None 1 = RADIUS 2 = LDAP
Authorization-Required	Y			66	Integer	Single	0 = No 1 = Yes
Authorization-DN-Field	Y	Y	Y	67	String	Single	Possible values: UID, OU, O, CN, L, SP, C, EA, T, N, GN, SN, I, GENQ, DNQ, SER, use-entire-name
IKE-KeepAlive-Confidence-Interval	Y	Y	Y	68	Integer	Single	10-300 seconds
WebVPN-Content-Filter-Parameters	Y	Y		69	Integer	Single	1 = Java ActiveX 2 = Java Script 4 = Image 8 = Cookies in images
WebVPN-URL-List		Y		71	String	Single	URL-List name
WebVPN-Port-Forward-List		Y		72	String	Single	Port-Forward list name
WebVPN-Access-List		Y		73	String	Single	Access-List name
Cisco-LEAP-Bypass	Y	Y	Y	75	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-Homepage	Y	Y		76	String	Single	A URL such as http://example-portal.com
Client-Type-Version-Limiting	Y	Y	Y	77	String	Single	IPSec VPN version number string

Table D-6 Security Appliance Supported RADIUS Attributes and Values (continued)

Attribute Name	VPN 3000	ASA	PIX	Attr. #	Syntax/ Type	Single or Multi-Valued	Description or Value
WebVPN-Port-Forwarding-Name	Y	Y		79	String	Single	String name (example, "Corporate-Apps"). This text replaces the default string, "Application Access," on the clientless portal home page.
IE-Proxy-Server	Y			80	String	Single	IP address
IE-Proxy-Server-Policy	Y			81	Integer	Single	1 = No Modify 2 = No Proxy 3 = Auto detect 4 = Use Concentrator Setting
IE-Proxy-Exception-List	Y			82	String	Single	newline (\n) separated list of DNS domains
IE-Proxy-Bypass-Local	Y			83	Integer	Single	0 = None 1 = Local
IKE-Keepalive-Retry-Interval	Y	Y	Y	84	Integer	Single	2 - 10 seconds
Tunnel-Group-Lock		Y	Y	85	String	Single	Name of the tunnel group or "none"
Access-List-Inbound		Y	Y	86	String	Single	Access list ID
Access-List-Outbound		Y	Y	87	String	Single	Access list ID
Perfect-Forward-Secrecy-Enable	Y	Y	Y	88	Boolean	Single	0 = No 1 = Yes
NAC-Enable	Y			89	Integer	Single	0 = No 1 = Yes
NAC-Status-Query-Timer	Y			90	Integer	Single	30 - 1800 seconds
NAC-Revalidation-Timer	Y			91	Integer	Single	300 - 86400 seconds
NAC-Default-ACL	Y			92	String		Access list
WebVPN-URL-Entry-Enable	Y	Y		93	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-File-Access-Enable	Y	Y		94	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-File-Server-Entry-Enable	Y	Y		95	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-File-Server-Browsing-Enable	Y	Y		96	Integer	Single	0 = Disabled 1 = Enabled

Table D-6 Security Appliance Supported RADIUS Attributes and Values (continued)

Attribute Name	VPN 3000	ASA	PIX	Attr. #	Syntax/ Type	Single or Multi-Valued	Description or Value
WebVPN-Port-Forwarding-Enable	Y	Y		97	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-Outlook-Exchange-Proxy-Enable	Y	Y		98	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-Port-Forwarding-HTTP-Proxy	Y	Y		99	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-Auto-Applet-Download-Enable	Y	Y		100	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-Citrix-Metaframe-Enable	Y	Y		101	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-Apply-ACL	Y	Y		102	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-SSL-VPN-Client-Enable	Y	Y		103	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-SSL-VPN-Client-Required	Y	Y		104	Integer	Single	0 = Disabled 1 = Enabled
WebVPN-SSL-VPN-Client-Keep-Installation	Y	Y		105	Integer	Single	0 = Disabled 1 = Enabled
SVC-Keepalive	Y	Y		107	Integer	Single	0 = Off 15 - 600 seconds
SVC-DPD-Interval-Client	Y	Y		108	Integer	Single	0 = Off 5 - 3600 seconds
SVC-DPD-Interval-Gateway	Y	Y		109	Integer	Single	0 = Off) 5 - 3600 seconds
SVC-Rekey-Time		Y		110	Integer	Single	0 = Disabled 1- 10080 minutes
WebVPN-Deny-Message		Y		116	String	Single	Valid string(up to 500 characters)
SVC-DTLS		Y		123	Integer	Single	0 = False 1 = True
SVC-MTU		Y		125	Integer	Single	MTU value 256 - 1406 in bytes
SVC-Modules		Y		127	String	Single	String (name of a module)
SVC-Profiles		Y		128	String	Single	String (name of a profile)

Table D-6 Security Appliance Supported RADIUS Attributes and Values (continued)

Attribute Name	VPN 3000	ASA	PIX	Attr. #	Syntax/ Type	Single or Multi-Valued	Description or Value
SVC-Ask		Y		131	String	Single	0 = Disabled 1 = Enabled 3 = Enable default service 5 = Enable default clientless (2 and 4 not used)
SVC-Ask-Timeout		Y		132	Integer	Single	5 - 120 seconds
IE-Proxy-PAC-URL		Y		133	String	Single	PAC Address String
Strip-Realm	Y	Y	Y	135	Boolean	Single	0 = Disabled 1 = Enabled
Smart-Tunnel		Y		136	String	Single	Name of a Smart Tunnel
WebVPN-ActiveX-Relay		Y		137	Integer	Single	0 = Disabled Otherwise = Enabled
Smart-Tunnel-Auto		Y		138	Integer	Single	0 = Disabled 1 = Enabled 2 = AutoStart
VLAN		Y		140	Integer	Single	0 - 4094
NAC-Settings		Y		141	String	Single	Name of NAC policy
Member-Of		Y	Y	145	String	Single	Comma delimited string, for example: `Engineering, Sales`
Address-Pools		Y	Y	217	String	Single	Name of IP local pool
IPv6-Address-Pools		Y		218	String	Single	Name of IP local pool-IPv6
IPv6-VPN-Filter		Y		219	String	Single	ACL value
Privilege-Level		Y	Y	220	Integer	Single	An integer between 0 and 15.
WebVPN-Macro-Value1		Y		223	String	Single	Unbounded
WebVPN-Macro-Value2		Y		224	String	Single	Unbounded

Configuring an External TACACS+ Server

The security appliance provides support for TACACS+ attributes. TACACS+ separates the functions of authentication, authorization, and accounting. The protocol supports two types of attributes: mandatory and optional. Both the server and client must understand a mandatory attribute, and the mandatory attribute must be applied to the user. An optional attribute may or may not be understood or used.

Note To use TACACS+ attributes, make sure you have enabled AAA services on the NAS.

Table D-7 lists supported TACACS+ authorization response attributes for cut-through-proxy connections. Table D-8 lists supported TACACS+ accounting attributes.

Table D-7 Supported TACACS+ Authorization Response Attributes

Attribute	Description
acl	Identifies a locally configured access list to be applied to the connection.
idletime	Indicates the amount of inactivity in minutes that is allowed before the authenticated user session is terminated.
timeout	Specifies the absolute amount of time in minutes that authentication credentials remain active before the authenticated user session is terminated.

Table D-8 Supported TACACS+ Accounting Attributes

Attribute	Description
bytes_in	Specifies the number of input bytes transferred during this connection (stop records only).
bytes_out	Specifies the number of output bytes transferred during this connection (stop records only).
cmd	Defines the command executed (command accounting only).
disc-cause	Indicates the numeric code that identifies the reason for disconnecting (stop records only).
elapsed_time	Defines the elapsed time in seconds for the connection (stop records only).
foreign_ip	Specifies the IP address of the client for tunnel connections. Defines the address on the lowest security interface for cut-through-proxy connections.
local_ip	Specifies the IP address that the client connected to for tunnel connections. Defines the address on the highest security interface for cut-through-proxy connections.
NAS port	Contains a session ID for the connection.
packs_in	Specifies the number of input packets transferred during this connection.
packs_out	Specifies the number of output packets transferred during this connection.
priv-level	Set to the user's privilege level for command accounting requests or to 1 otherwise.
rem_iddr	Indicates the IP address of the client.
service	Specifies the service used. Always set to "shell" for command accounting only.
task_id	Specifies a unique task ID for the accounting transaction.
username	Indicates the name of the user.



APPENDIX E

Configuring the Security Appliance for Use with MARS

MARS centrally aggregates logs and events from various network devices, including security appliances, which you can analyze for use in threat mitigation. MARS supports the following PIX and ASA adaptive security appliance versions: 7.0(7), 7.2(2), 7.2(3), and 8.0(2).

This appendix describes how to configure the security appliance and add it to MARS as a reporting device, and includes the following sections:

- Taskflow for Configuring MARS to Monitor Security Appliances, page E-1
- Enabling Administrative Access to MARS on the Security Appliance, page E-2
- Adding a Security Appliance to Monitor, page E-3
- Setting the Logging Severity Level for System Log Messages, page E-5
- System Log Messages That Are Processed by MARS, page E-5
- Configuring Specific Features, page E-7

For more information about configuring devices and software to work with MARS, see the *Supported and Interoperable Devices and Software for Cisco Security MARS Local Controller* document and the *User Guide for Cisco Security MARS Local Controller*.

Taskflow for Configuring MARS to Monitor Security Appliances

The taskflow for configuring MARS to monitor the security appliance includes the following steps:

1. Configure the security appliance to accept administrative sessions from MARS to discover settings. Configure this setting in the admin context.
2. Configure the security appliance to publish its system log messages to MARS. Configure this setting for the admin context and for each security context defined.

   Note Each context requires a unique, routable IP address for sending system log messages to MARS, and each context must have a unique name (usually in the *hostname.domain* name format).

3. To enable MARS to accept system log message event data and to collect configuration settings from the security appliance, perform the following tasks:
   - Enable logging for one or more interfaces.

- Select the logging facility and queue size.
- Specify the logging severity level as debugging (7) or indicate the desired severity level.
- Identify the target MARS appliance, and the protocol and port pair on which it listens.

4. Within the MARS web interface, perform the following steps:
   - Define the security appliance by providing the administrative connection information.
   - Define security contexts. For more information, see Adding Security Contexts, page E-4.
   - Add discovered contexts. For more information, see Adding Discovered Contexts, page E-4.
   - Edit discovered contexts. For more information, see Editing Discovered Contexts, page E-5.

Enabling Administrative Access to MARS on the Security Appliance

To enable administrative access to MARS on the security appliance, perform the following steps:

Step 1 To enable the MARS appliance to discover the security appliance settings through SSH access, enter the following commands:

```
hostname# crypto key generate rsa modulus modulus
```

where *modulus* is the RSA modulus size specified in bits

```
hostname# ssh mars_ip netmask of the mars_ip interface name
```

where *mars_ip* is the IP address of the MARS appliance, *netmask of the mars_ip* is the netmask of the MARS appliance, and *interface name* can be inside, outside, or DMZ.

Step 2 To enable the MARS appliance to discover the security appliance settings through Telnet access, enter the following command:

```
hostname# telnet mars_ip netmask of the mars_ip interface name
```

where *mars_ip* is the IP address of the MARS appliance, *netmask of the mars_ip* is the netmask of the MARS appliance, and *interface name* can be inside, outside, or DMZ.

Step 3 To enable the MARS appliance to discover the security appliance settings through FTP access, make sure that you have added the MARS appliance configuration file to an FTP server.

Note If you choose the FTP access type, the MARS appliance cannot discover the non-admin context settings. Therefore, we do not recommend this access type.

Step 4 To enable MARS to act as a target logging host, configure the security appliance to publish system log messages to MARS by entering the following commands:

```
hostname# logging host interface name mars_ip
```

where *mars_ip* is the IP address of the MARS appliance and *interface name* can be inside, outside, or DMZ.

```
hostname# logging trap 7
```

```
hostname# logging enable
```

Note Make sure that you set the logging severity level to 7 (debugging), or configure the security appliance to generate the desired set of system log messages. The logging severity level generates the system log message details that are required to track session-specific data.

Debugging messages are recommended for troubleshooting. The debugging logging severity level includes all emergency, alert, critical, error, warning, notification, and informational messages. This logging severity level also generates logs that identify the commands that are issued during FTP sessions and the URLs that are requested during HTTP sessions. If the security appliance cannot sustain debugging-level messages because of performance considerations, use the informational logging severity level (6). For more information, see Setting the Logging Severity Level for System Log Messages, page E-5.

In addition, do not use the EMBLEM format for system log messages.

Step 5 To allow MARS to discover CPU usage and related information, enable the SNMP RO community string for the security appliance by entering the following command:

```
hostname# snmp-server host interface mars_ip poll community community
```

where *interface* can be inside, outside, or DMZ, *mars_ip* is the IP address of the MARS appliance, and *community* is the SNMP RO community string.

Step 6 Repeat Step 4 for each admin context and security context defined.

Adding a Security Appliance to Monitor

Events that are published by a reporting device (the security appliance) to MARS are not inspected until the reporting IP address of the security appliance is defined in the MARS web interface.

To add a PIX or ASA adaptive security appliance to monitor, perform the following steps:

Step 1 In the MARS web interface, click **Admin > System Setup > Security and Monitor Devices > Add**.

Step 2 Choose the correct version of the ASA adaptive security appliance from the Device Type drop-down list. The basic device type represents the admin context.

Step 3 Specify values for the following Device Access fields:

Tip To enable SSH discovery, the MARS appliance must authenticate to the security appliance. The default username is “pix” and the password is the one that you specified for the **password** command (unless you use AAA).

- Device Name, which MARS maps to the reporting IP address
- Access IP, which is usually the same as the reporting IP address
- Reporting IP, which is the interface that publishes system log messages or SNMP notifications, or both
- Access Type
- Login

- Password
- Enable Password
- (Optional) SNMP RO, which allows MARS to retrieve MIBs that are related to CPU usage and network usage
- (Optional) Monitor Resource Usage (requires the SNMP RO setting), which allows MARS to monitor for anomalous consumption of resources, such as memory and CPU

Step 4 Click **Discover** to determine the security appliance settings, including any security contexts and their settings.

Step 5 Click **Submit** to save these settings in the MARS database.

Step 6 Click **Activate to load these settings into the MARS appliance working memory**.

Step 7 Click **Summary > Dashboard**.

Step 8 Under the Hotspot Graph, click **Full Topology Graph**, and verify that the selected security appliance appears.

Adding Security Contexts

To add security contexts, perform the following steps:

Step 1 In the MARS web interface, click **Add Module**.

Step 2 Choose the correct version of the security appliance from the Device Type drop-down list.

Step 3 Enter the name of the security appliance in the Device Name field.

Step 4 Enter the name of the security context in the Context Name field. This name must match the context name defined on the security appliance.

Step 5 Enter the IP address of the security context from which system log messages or SNMP notifications, or both are published in the Reporting IP field.

Step 6 (Optional) Enter the security appliance read-only community string in the SNMP RO Community field.

Step 7 Click **Discover** to discover the settings of the defined security context. MARS collects all route, NAT, and ACL-related information.

Step 8 Click **Submit** to save these settings in the MARS database.

Adding Discovered Contexts

To add discovered contexts, perform the following steps:

Step 1 In the MARS web interface, click **Add Available Module**.

Step 2 Choose the security context from the Select drop-down list, and click **Add**.

Step 3 Click **Submit** to save these settings in the MARS database.

Step 4 Repeat these steps for each discovered context.

Editing Discovered Contexts

To edit discovered contexts, perform the following steps:

Step 1 In the MARS web interface, choose the discovered context that you want to edit according to the selected device type.

Step 2 Click **Edit Module**.

Step 3 Enter the IP address from which the system log messages of the security context are sent in the Reporting IP field.

Step 4 (Optional) Enter the security appliance read-only community string in the SNMP RO Community field.

Step 5 (Optional) To enable MARS to monitor this context for anomalous resource usage, click **Yes** from the Monitor Resource Usage list.

Step 6 Click **Submit** to save these settings in the MARS database.

Step 7 Repeat these steps for each discovered context.

Setting the Logging Severity Level for System Log Messages

You can change the logging severity level of the required system log messages or turn off specific system log messages using the **logging message** command. For more information, see Chapter 44, "Monitoring the Security Appliance."

System Log Messages That Are Processed by MARS

MARS can correctly parse system log messages at customized logging severity levels. Therefore, you can set system log messages to a lower logging severity level (for example, logging severity level 6). By changing the logging severity level for system log messages, you can reduce the logging load on the security appliance by 5-15%. However, the primary consumer of resources are the session detail events.

MARS processes the following system log messages, which are required for correct sessionization. If you change the logging severity level of the security appliance, make sure that these system log messages are generated at the new logging severity level so that the MARS appliance can receive them.

Table E-1 lists the system log message classes, their definitions, and the ranges of system log message numbers that are processed by MARS.

Table E-1 System Log Message Classes and Associated Message Numbers

Class	Definition	System Log Message Numbers
auth	User Authentication	109001-109003, 109005-109008, 109010-109014, 109016-109034, 113001, 113003-113020, 114001-114020, 611101-611104, 611301-611323
bridge	Transparent Firewall	110001
ca	PKI Certification Authority	717001-717019, 717021-717038

Table E-1 System Log Message Classes and Associated Message Numbers (continued)

Class (continued)	Definition	System Log Message Numbers
config	Command Interface	111001, 111003-111005, 111007-111009, 111111, 112001, 208005, 308001-308002, 504001-504002, 505001-505013, 506001
e-mail	E-mail Proxy	719001-719026
ha	High Availability (Failover)	101001-101005, 102001, 103001-103005, 104001-104004, 105001-105011, 105020-105021, 105031-105032, 105034-105040, 105042-105048, 210001-210003, 210005-210008, 210010, 210020-210022, 311001-311004, 709001-709007
ip	IP Stack	209003-209005, 215001, 313001, 313003-313005, 313008, 317001-317005, 322001-322004, 323001-323006, 324000-324007, 324300-324301, 325001-325003, 326001-326002, 326004-326017, 326019-326028, 327001-327003, 328001, 329001, 331001-331002, 332003-332004, 333001-333010, 334001-334008, 335001-335014, 408001-408003, 410001-410004, 411001-411004, 412001-412002, 413001-413004, 416001, 417001, 417004, 417006, 417008-417009, 418001, 419001-419002, 421001-421007, 422004-422006, 423001-423005, 424001-424002, 431001-431002, 450001, 507001-507002, 508001-508002, 509001
ips	Intrusion Protection Service	400000-400050, 401001-401005, 415001-415020, 420001-420003
np	Network Processor	319001-319004
npssl	NP SSL	725001-725014
ospf	OSPF Routing	318001-318009, 409001-409013, 409023, 503001, 613001-613003
rip	RIP Routing	107001-107003, 312001
rm	Resource Manager	321001-321004
session	User Session	106001-106002, 106006-106007, 106010-106027, 106100-106101, 108002-108003, 108005, 201002-201006, 201008-201013, 202001, 201005, 202011, 204001, 302001, 302003-302004, 302007-302010, 302012-302023, 302302, 303002-303005, 304001-304009, 305005-305012, 314001, 405001-405002, 405101-405107, 405201, 405300-405301, 406001-406002, 407001-407003, 500001-500004, 502101-502103, 502111-502112, 607001-607002, 608001-608005, 609001-609002, 616001, 617001-617004, 620001-620002, 621001-621003, 621006-621010, 622001, 622101-622102, 703001-703002, 710001-710006, 726001
snmp	SNMP	212001-212006

Table E-1 System Log Message Classes and Associated Message Numbers (continued)

Class (continued)	Definition	System Log Message Numbers
sys	System	199001-199003, 199005-199009, 211001, 211003, 216003, 217001, 218001-218004, 219002, 315004, 315011, 414001-414002, 604101-604104, 605004-605005, 606001-606004, 610001-610002, 610101, 612001-612003, 614001-614002, 615001-615002, 701001-701002, 711001-711002
vpdn	PPTP and L2TP Sessions	213001-213004, 403101-403104, 403106-403110, 403500-403507, 603101-603109
vpn	IKE and IPSec	316001, 320001, 402101-402103, 402106, 402114-402120, 402123, 404101-404102, 501101, 602101-602104, 602201-602203, 602301-602304, 702201-702212, 702301-702303, 702305, 702307, 713004, 713006, 713008-713010, 713012, 713014, 713016-713018, 713020, 713022, 713024-713037, 713039-713043, 713047-713052, 713056, 713059-713063, 713065-713066, 713068, 713072-713076, 713078, 713081-713086, 713088, 713092, 713094, 713098-713099, 713102-713105, 713107, 713109, 713112-713124, 713127-713149, 713152, 713154-713172, 713174, 713176-713179, 713182, 713184-713187, 713189-713190, 713193-713199, 713203-713206, 713208-713226, 713228-713251, 713900-713906, 714001-714007, 714011, 715001, 715004-715009, 715013, 715019-715022, 715027-715028, 715033-715042, 715044-715072, 715074-715079
vpnc	VPN Client	611101-611104, 611301-611323, 722001-722038
vpnfo	VPN Failover	720001-720073
vpnlb	VPN Load Balancing	718001-718081, 718084-718088
webvpn	Web-based VPN	716001-716056, 723001-723014, 724001-724002

Configuring Specific Features

You can configure security appliances to act as reporting devices and manual mitigation devices, because they perform multiple roles on your network. MARS can benefit from configuration of the following features:

- The built-in IDS and IPS signature matching features can be critical in detecting an attempted attack.
- The logging of accepted, as well as denied sessions, aids in false positive analysis.
- Administrative access ensures that MARS can obtain critical data, including the following:
  - *Route and ARP tables*, which aid in network discovery and MAC address mapping.
  - *NAT and PAT translation tables*, which aid in address resolution and attack path analysis, and expose the actual instigator of attacks.
  - *OS settings*, from which MARS determines the correct ACLs to block detected attacks, which you can use in a management session with the security appliance.



GLOSSARY

Numerics | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X

Numerics

3DES See DES.

A

AAA Authentication, authorization, and accounting. See also TACACS+ and RADIUS.

ABR Area Border Router. In OSPF, a router with interfaces in multiple areas.

ACE Access Control Entry. Information entered into the configuration that lets you specify what type of traffic to permit or deny on an interface. By default, traffic that is not explicitly permitted is denied.

Access Modes The security appliance CLI uses several command modes. The commands available in each mode vary. See also user EXEC mode, privileged EXEC mode, global configuration mode, command-specific configuration mode.

ACL Access Control List. A collection of ACEs. An ACL lets you specify what type of traffic to allow on an interface. By default, traffic that is not explicitly permitted is denied. ACLs are usually applied to the interface which is the source of inbound traffic. See also rule, outbound ACL.

ActiveX A set of object-oriented programming technologies and tools used to create mobile or portable programs. An ActiveX program is roughly equivalent to a Java applet.

Address Resolution Protocol See ARP.

address translation The translation of a network address and/or port to another network address/or port. See also IP address, interface PAT, NAT, PAT, Static PAT, xlate.

AES Advanced Encryption Standard. A symmetric block cipher that can encrypt and decrypt information. The AES algorithm is capable of using cryptographic keys of 128, 192 and 256 bits to encrypt and decrypt data in blocks of 128 bits. See also DES.

AH Authentication Header. An IP protocol (type 51) that can ensure data integrity, authentication, and replay detection. AH is embedded in the data to be protected (a full IP datagram, for example). AH can be used either by itself or with ESP. This is an older IPSec protocol that is less important in most networks than ESP. AH provides authentication services but does not provide encryption services. It is provided to ensure compatibility with IPSec peers that do not support ESP, which provides both authentication and encryption. See also encryption and VPN. Refer to the RFC 2402.

A record address "A" stands for address, and refers to name-to-address mapped records in DNS.

APCF Application Profile Customization Framework. Lets the security appliance handle non-standard applications so that they render correctly over a WebVPN connection.

ARP Address Resolution Protocol. A low-level TCP/IP protocol that maps a hardware address, or MAC address, to an IP address. An example hardware address is 00:00:a6:00:01:ba. The first three groups of characters (00:00:a6) identify the manufacturer; the rest of the characters (00:01:ba) identify the system card. ARP is defined in RFC 826.

ASA Adaptive Security Algorithm. Used by the security appliance to perform inspections. ASA allows one-way (inside to outside) connections without an explicit configuration for each internal system and application. See also inspection engine.

ASA adaptive security appliance.

ASDM Adaptive Security Device Manager. An application for managing and configuring a single security appliance.

asymmetric encryption Also called public key systems, asymmetric encryption allows anyone to obtain access to the public key of anyone else. Once the public key is accessed, one can send an encrypted message to that person using the public key. See also encryption, public key.

authentication Cryptographic protocols and services that verify the identity of users and the integrity of data. One of the functions of the IPSec framework. Authentication establishes the integrity of datastream and ensures that it is not tampered with in transit. It also provides confirmation about the origin of the datastream. See also AAA, encryption, and VPN.

Auto Applet Download Automatically downloads the WebVPN port-forwarding applet when the user first logs in to WebVPN.

auto-signon This command provides a single sign-on method for WebVPN users. It passes the WebVPN login credentials (username and password) to internal servers for authentication using NTLM authentication, basic authentication, or both.

B

Backup Server IPSec backup servers let a VPN client connect to the central site when the primary security appliance is unavailable.

BGP Border Gateway Protocol. BGP performs interdomain routing in TCP/IP networks. BGP is an Exterior Gateway Protocol, which means that it performs routing between multiple autonomous systems or domains and exchanges routing and access information with other BGP systems. The security appliance does not support BGP. See also EGP.

BLT stream Bandwidth Limited Traffic stream. Stream or flow of packets whose bandwidth is constrained.

BOOTP Bootstrap Protocol. Lets diskless workstations boot over the network as is described in RFC 951 and RFC 1542.

BPDU Bridge Protocol Data Unit. Spanning-Tree Protocol hello packet that is sent out at configurable intervals to exchange information among bridges in the network. Protocol data unit is the OSI term for packet.

C

CA Certificate Authority, Certification Authority. A third-party entity that is responsible for issuing and revoking certificates. Each device with the public key of the CA can authenticate a device that has a certificate issued by the CA. The term CA also refers to software that provides CA services. See also certificate, CRL, public key, RA.

cache A temporary repository of information accumulated from previous task executions that can be reused, decreasing the time required to perform the tasks. Caching stores frequently reused objects in the system cache, which reduces the need to perform repeated rewriting and compressing of content.

CBC Cipher Block Chaining. A cryptographic technique that increases the encryption strength of an algorithm. CBC requires an initialization vector (IV) to start encryption. The IV is explicitly given in the IPSec packet.

certificate A signed cryptographic object that contains the identity of a user or device and the public key of the CA that issued the certificate. Certificates have an expiration date and may also be placed on a CRL if known to be compromised. Certificates also establish non-repudiation for IKE negotiation, which means that you can prove to a third party that IKE negotiation was completed with a specific peer.

CHAP Challenge Handshake Authentication Protocol.

CIFS Common Internet File System. It is a platform-independent file sharing system that provides users with network access to files, printers, and other machine resources. Microsoft implemented CIFS for networks of Windows computers, however, open source implementations of CIFS provide file access to servers running other operating systems, such as Linux, UNIX, and Mac OS X.

Citrix An application that virtualizes client-server applications and optimizes web applications.

CLI command line interface. The primary interface for entering configuration and monitoring commands to the security appliance.

client/server computing Distributed computing (processing) network systems in which transaction responsibilities are divided into two parts: client (front end) and server (back end). Also called distributed computing. See also RPC.

Client update Lets you update revisions of clients to which the update applies; provide a URL or IP address from which to get the update; and, in the case of Windows clients, optionally notify users that they should update their VPN client version.

command-specific configuration mode From global configuration mode, some commands enter a command-specific configuration mode. All user EXEC, privileged EXEC, global configuration, and command-specific configuration commands are available in this mode. See also global configuration mode, privileged EXEC mode, user EXEC mode.

Compression The process of encoding information using fewer bits or other information-bearing units than an unencoded representation would use. Compression can reduce the size of transferring packets and increase communication performance.

configuration, config, config file A file on the security appliance that represents the equivalent of settings, preferences, and properties administered by ASDM or the CLI.

Content Rewriting/Transformation Interprets and modifies applications so that they render correctly over a WebVPN connection.

cookie A cookie is a object stored by a browser. Cookies contain information, such as user preferences, to persistent storage.

CPU Central Processing Unit. Main processor.

CRC Cyclical Redundancy Check. Error-checking technique in which the frame recipient calculates a remainder by dividing frame contents by a prime binary divisor and compares the calculated remainder to a value stored in the frame by the sending node.

CRL Certificate Revocation List. A digitally signed message that lists all of the current but revoked certificates listed by a given CA. This is analogous to a book of stolen charge card numbers that allow stores to reject bad credit cards. When certificates are revoked, they are added to a CRL. When you implement authentication using certificates, you can choose to use CRLs or not. Using CRLs lets you easily revoke certificates before they expire, but the CRL is generally only maintained by the CA or an RA. If you are using CRLs and the connection to the CA or RA is not available when authentication is requested, the authentication request will fail. See also CA, certificate, public key, RA.

CRV Call Reference Value. Used by H.225.0 to distinguish call legs signalled between two entities.

cryptography Encryption, authentication, integrity, keys and other services used for secure communication over networks. See also VPN and IPSec.

crypto map A data structure with a unique name and sequence number that is used for configuring VPNs on the security appliance. A crypto map selects data flows that need security processing and defines the policy for these flows and the crypto peer that traffic needs to go to. A crypto map is applied to an interface. Crypto maps contain the ACLs, encryption standards, peers, and other parameters necessary to specify security policies for VPNs using IKE and IPSec. See also VPN.

CTIQBE Computer Telephony Interface Quick Buffer Encoding. A protocol used in IP telephony between the Cisco CallManager and CTI TAPI and JTAPI applications. CTIQBE is used by the TAPI/JTAPI protocol inspection module and supports NAT, PAT, and bi-directional NAT. This enables Cisco IP SoftPhone and other Cisco TAPI/JTAPI applications to communicate with Cisco CallManager for call setup and voice traffic across the security appliance.

cut-through proxy Enables the security appliance to provide faster traffic flow after user authentication. The cut-through proxy challenges a user initially at the application layer. After the security appliance authenticates the user, it shifts the session flow and all traffic flows directly and quickly between the source and destination while maintaining session state information.

D

data confidentiality Describes any method that manipulates data so that no attacker can read it. This is commonly achieved by data encryption and keys that are only available to the parties involved in the communication.

data integrity Describes mechanisms that, through the use of encryption based on secret key or public key algorithms, allow the recipient of a piece of protected data to verify that the data has not been modified in transit.

data origin authentication A security service where the receiver can verify that protected data could have originated only from the sender. This service requires a data integrity service plus a key distribution mechanism, where a secret key is shared only between the sender and receiver.

decryption Application of a specific algorithm or cipher to encrypted data so as to render the data comprehensible to those who are authorized to see the information. See also encryption.

DES Data encryption standard. DES was published in 1977 by the National Bureau of Standards and is a secret key encryption scheme based on the Lucifer algorithm from IBM. Cisco uses DES in classic crypto (40-bit and 56-bit key lengths), IPSec crypto (56-bit key), and 3DES (triple DES), which performs encryption three times using a 56-bit key. 3DES is more secure than DES but requires more processing for encryption and decryption. See also AES, ESP.

DHCP Dynamic Host Configuration Protocol. Provides a mechanism for allocating IP addresses to hosts dynamically, so that addresses can be reused when hosts no longer need them and so that mobile computers, such as laptops, receive an IP address applicable to the LAN to which it is connected.

Diffie-Hellman A public key cryptography protocol that allows two parties to establish a shared secret over insecure communications channels. Diffie-Hellman is used within IKE to establish session keys. Diffie-Hellman is a component of Oakley key exchange.

Diffie-Hellman Group 1, Group 2, Group 5, Group 7 Diffie-Hellman refers to a type of public key cryptography using asymmetric encryption based on large prime numbers to establish both Phase 1 and Phase 2 SAs. Group 1 provides a smaller prime number than Group 2 but may be the only version supported by some IPSec peers. Diffe-Hellman Group 5 uses a 1536-bit prime number, is the most secure, and is recommended for use with AES. Group 7 has an elliptical curve field size of 163 bits and is for use with the Movian VPN client, but works with any peer that supports Group 7 (ECC). See also VPN and encryption.

Note The **group 7** command option was **deprecated** in ASA version 8.0(4).

digital certificate See certificate.

DMZ See interface.

DN Distinguished Name. Global, authoritative name of an entry in the OSI Directory (X.500).

DNS Domain Name System (or Service). An Internet service that translates domain names into IP addresses.

DoS Denial of Service. A type of network attack in which the goal is to render a network service unavailable.

DSL digital subscriber line. Public network technology that delivers high bandwidth over conventional copper wiring at limited distances. DSL is provisioned via modem pairs, with one modem located at a central office and the other at the customer site. Because most DSL technologies do not use the whole bandwidth of the twisted pair, there is room remaining for a voice channel.

DSP digital signal processor. A DSP segments a voice signal into frames and stores them in voice packets.

DSS Digital Signature Standard. A digital signature algorithm designed by The US National Institute of Standards and Technology and based on public-key cryptography. DSS does not do user datagram encryption. DSS is a component in classic crypto, as well as the Redcreek IPSec card, but not in IPSec implemented in Cisco IOS software.

Dynamic NAT See NAT and address translation.

Dynamic PAT Dynamic Port Address Translation. Dynamic PAT lets multiple outbound sessions appear to originate from a single IP address. With PAT enabled, the security appliance chooses a unique port number from the PAT IP address for each outbound translation slot (xlate). This feature is valuable when an ISP cannot allocate enough unique IP addresses for your outbound connections. The global pool addresses always come first, before a PAT address is used. See also NAT, Static PAT, and xlate.

E

ECHO See Ping, ICMP. See also inspection engine.

EGP Exterior Gateway Protocol. Replaced by BGP. The security appliance does not support EGP. See also BGP.

EIGRP Enhanced Interior Gateway Routing Protocol. The security appliance does not support EIGRP.

EMBLEM Enterprise Management BaseLine Embedded Manageability. A syslog format designed to be consistent with the Cisco IOS system log format and is more compatible with CiscoWorks management applications.

encryption Application of a specific algorithm or cipher to data so as to render the data incomprehensible to those unauthorized to see the information. See also decryption.

ESMTP Extended SMTP. Extended version of SMTP that includes additional functionality, such as delivery notification and session delivery. ESMTP is described in RFC 1869, SMTP Service Extensions.

ESP Encapsulating Security Payload. An IPSec protocol, ESP provides authentication and encryption services for establishing a secure tunnel over an insecure network. For more information, refer to RFCs 2406 and 1827.

F

failover, failover mode Failover lets you configure two security appliances so that one will take over operation if the other one fails. The security appliance supports two failover configurations, Active/Active failover and Active/Standby failover. Each failover configuration has its own method for determining and performing failover. With Active/Active failover, both units can pass network traffic. This lets you configure load balancing on your network. Active/Active failover is only available on units running in multiple context mode. With Active/Standby failover, only one unit passes traffic while the other unit waits in a standby state. Active/Standby failover is available on units running in either single or multiple context mode.

Fixup See inspection engine.

Flash, Flash memory A nonvolatile storage device used to store the configuration file when the security appliance is powered down.

FQDN/IP Fully qualified domain name/IP address. IPSec parameter that identifies peers that are security gateways.

FragGuard Provides IP fragment protection and performs full reassembly of all ICMP error messages and virtual reassembly of the remaining IP fragments that are routed through the security appliance.

FTP File Transfer Protocol. Part of the TCP/IP protocol stack, used for transferring files between hosts.

G

GGSN gateway GPRS support node. A wireless gateway that allows mobile cell phone users to access the public data network or specified private IP networks.

global configuration mode Global configuration mode lets you to change the security appliance configuration. All user EXEC, privileged EXEC, and global configuration commands are available in this mode. See also user EXEC mode, privileged EXEC mode, command-specific configuration mode.

GMT Greenwich Mean Time. Replaced by UTC (Coordinated Universal Time) in 1967 as the world time standard.

GPRS general packet radio service. A service defined and standardized by the European Telecommunication Standards Institute. GPRS is an IP-packet-based extension of GSM networks and provides mobile, wireless, data communications

GRE Generic Routing Encapsulation described in RFCs 1701 and 1702. GRE is a tunneling protocol that can encapsulate a wide variety of protocol packet types inside IP tunnels, creating a virtual point-to-point link to routers at remote points over an IP network. By connecting multiprotocol subnetworks in a single-protocol backbone environment, IP tunneling using GRE allows network expansion across a single protocol backbone environment.

GSM Global System for Mobile Communication. A digital, mobile, radio standard developed for mobile, wireless, voice communications.

GTP GPRS tunneling protocol. GTP handles the flow of user packet data and signaling information between the SGSN and GGSN in a GPRS network. GTP is defined on both the Gn and Gp interfaces of a GPRS network.

H

H.225 A protocol used for TCP signalling in applications such as video conferencing. See also H.323 and inspection engine.

H.225.0 An ITU standard that governs H.225.0 session establishment and packetization. H.225.0 actually describes several different protocols: RAS, use of Q.931, and use of RTP.

H.245 An ITU standard that governs H.245 endpoint control.

H.320 Suite of ITU-T standard specifications for video conferencing over circuit-switched media, such as ISDN, fractional T-1, and switched-56 lines. Extensions of ITU-T standard H.320 enable video conferencing over LANs and other packet-switched networks, as well as video over the Internet.

H.323 Allows dissimilar communication devices to communicate with each other by using a standardized communication protocol. H.323 defines a common set of CODECs, call setup and negotiating procedures, and basic data transport methods.

H.323 RAS Registration, admission, and status signaling protocol. Enables devices to perform registration, admissions, bandwidth changes, and status and disengage procedures between VoIP gateway and the gatekeeper.

H.450.2 Call transfer supplementary service for H.323.

H.450.3 Call diversion supplementary service for H.323.

Hash, Hash Algorithm A hash algorithm is a one way function that operates on a message of arbitrary length to create a fixed-length message digest used by cryptographic services to ensure its data integrity. MD5 has a smaller digest and is considered to be slightly faster than SHA-1. Cisco uses both SHA-1 and MD5 hashes within our implementation of the IPSec framework. See also encryption, HMAC, and VPN.

headend A firewall, concentrator, or other host that serves as the entry point into a private network for VPN client connections over the public network. See also ISP and VPN.

HMAC A mechanism for message authentication using cryptographic hashes such as SHA-1 and MD5.

host The name for any device on a TCP/IP network that has an IP address. See also network and node.

host/network An IP address and netmask used with other information to identify a single host or network subnet for security appliance configuration, such as an address translation (xlate) or ACE.

HTTP Hypertext Transfer Protocol. A protocol used by browsers and web servers to transfer files. When a user views a web page, the browser can use HTTP to request and receive the files used by the web page. HTTP transmissions are not encrypted.

HTTPS Hypertext Transfer Protocol Secure. An SSL-encrypted version of HTTP.

I

IANA Internet Assigned Number Authority. Assigns all port and protocol numbers for use on the Internet.

ICMP Internet Control Message Protocol. Network-layer Internet protocol that reports errors and provides other information relevant to IP packet processing.

IDS Intrusion Detection System. A method of detecting malicious network activity by signatures and then implementing a policy for that signature.

IETF The Internet Engineering Task Force. A technical standards organization that develops RFC documents defining protocols for the Internet.

IGMP Internet Group Management Protocol. IGMP is a protocol used by IPv4 systems to report IP multicast memberships to neighboring multicast routers.

IKE Internet Key Exchange. IKE establishes a shared security policy and authenticates keys for services (such as IPSec) that require keys. Before any IPSec traffic can be passed, each security appliance must verify the identity of its peer. This can be done by manually entering preshared keys into both hosts or by a CA service. IKE is a hybrid protocol that uses part Oakley and part of another protocol suite called SKEME inside ISAKMP framework. This is the protocol formerly known as ISAKMP/Oakley, and is defined in RFC 2409.

IKE Extended Authentication IKE Extended Authenticate (Xauth) is implemented per the IETF draft-ietf-ipsec-isakmp-xauth-04.txt ("extended authentication" draft). This protocol provides the capability of authenticating a user within IKE using TACACS+ or RADIUS.

IKE Mode Configuration IKE Mode Configuration is implemented per the IETF draft-ietf-ipsec-isakmp-mode-cfg-04.txt. IKE Mode Configuration provides a method for a security gateway to download an IP address (and other network level configuration) to the VPN client as part of an IKE negotiation.

ILS Internet Locator Service. ILS is based on LDAP and is ILSv2 compliant. ILS was developed by Microsoft for use with its NetMeeting, SiteServer, and Active Directory products.

IMAP Internet Message Access Protocol. Method of accessing e-mail or bulletin board messages kept on a mail server that can be shared. IMAP permits client e-mail applications to access remote message stores as if they were local without actually transferring the message.

implicit rule An access rule automatically created by the security appliance based on default rules or as a result of user-defined rules.

IMSI International Mobile Subscriber Identity. One of two components of a GTP tunnel ID, the other being the NSAPI. See also NSAPI.

inside The first interface, usually port 1, that connects your internal, "trusted" network protected by the security appliance. See also interface, interface names.

inspection engine The security appliance inspects certain application-level protocols to identify the location of embedded addressing information in traffic. This allows NAT to translate these embedded addresses and to update any checksum or other fields that are affected by the translation. Because many protocols open secondary TCP or UDP ports, each application inspection engine also monitors sessions to determine the port numbers for secondary channels. The initial session on a well-known port is used to negotiate dynamically assigned port numbers. The application inspection engine monitors these sessions, identifies the dynamic port assignments, and permits data exchange on these ports for the duration of the specific session. Some of the protocols that the security appliance can inspect are CTIQBE, FTP, H.323, HTTP, MGCP, SMTP, and SNMP.

interface The physical connection between a particular network and a security appliance.

interface ip_address The IP address of a security appliance network interface. Each interface IP address must be unique. Two or more interfaces must not be given the same IP address or IP addresses that are on the same IP network.

interface names Human readable name assigned to a security appliance network interface. The inside interface default name is "inside" and the outside interface default name is "outside." Any perimeter interface default names are "intf*n*", such as intf2 for the first perimeter interface, intf3 for the second perimeter interface, and so on to the last interface. The numbers in the intf string corresponds to the position of the interface card in the security appliance. You can use the default names or, if you are an experienced user, give each interface a more meaningful name. See also inside, intfn, outside.

intf*n* Any interface, usually beginning with port 2, that connects to a subset network of your design that you can custom name and configure.

interface PAT The use of PAT where the PAT IP address is also the IP address of the outside interface. See Dynamic PAT, Static PAT.

Internet The global network that uses IP. Not a LAN. See also intranet.

intranet Intranetwork. A LAN that uses IP. See also network and Internet.

IP Internet Protocol. IP protocols are the most popular nonproprietary protocols because they can be used to communicate across any set of interconnected networks and are equally well suited for LAN and WAN communications.

IPS Intrusion Prevention Service. An in-line, deep-packet inspection-based solution that helps mitigate a wide range of network attacks.

IP address An IP protocol address. A security appliance interface ip_address. IP version 4 addresses are 32 bits in length. This address space is used to designate the network number, optional subnetwork number, and a host number. The 32 bits are grouped into four octets (8 binary bits), represented by 4 decimal numbers separated by periods, or dots. The meaning of each of the four octets is determined by their use in a particular network.

IP pool A range of local IP addresses specified by a name, and a range with a starting IP address and an ending address. IP Pools are used by DHCP and VPNs to assign local IP addresses to clients on the inside interface.

IPSec IP Security. A framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer. IPSec uses IKE to handle the negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPSec. IPSec can protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

IPSec Phase 1 The first phase of negotiating IPSec, includes the key exchange and the ISAKMP portions of IPSec.

IPSec Phase 2 The second phase of negotiating IPSec. Phase two determines the type of encryption rules used for payload, the source and destination that will be used for encryption, the definition of interesting traffic according to access lists, and the IPSec peer. IPSec is applied to the interface in Phase 2.

IPSec transform set A transform set specifies the IPSec protocol, encryption algorithm, and hash algorithm to use on traffic matching the IPSec policy. A transform describes a security protocol (AH or ESP) with its corresponding algorithms. The IPSec protocol used in almost all transform sets is ESP with the DES algorithm and HMAC-SHA for authentication.

ISAKMP Internet Security Association and Key Management Protocol. A protocol framework that defines payload formats, the mechanics of implementing a key exchange protocol, and the negotiation of a security association. See IKE.

ISP Internet Service Provider. An organization that provides connection to the Internet via their services, such as modem dial in over telephone voice lines or DSL.

J

JTAPI Java Telephony Application Programming Interface. A Java-based API supporting telephony functions. See also TAPI.

K

key A data object used for encryption, decryption, or authentication.

L

LAN Local area network. A network residing in one location, such as a single building or campus. See also Internet, intranet, and network.

layer, layers Networking models implement layers with which different protocols are associated. The most common networking model is the OSI model, which consists of the following 7 layers, in order: physical, data link, network, transport, session, presentation, and application.

LCN Logical channel number.

LDAP Lightweight Directory Access Protocol. LDAP provides management and browser applications with access to X.500 directories.

M

mask A 32-bit mask that shows how an Internet address is divided into network, subnet, and host parts. The mask has ones in the bit positions to be used for the network and subnet parts, and zeros for the host part. The mask should contain at least the standard network portion, and the subnet field should be contiguous with the network portion.

MCR See multicast.

MC router Multicast (MC) routers route multicast data transmissions to the hosts on each LAN in an internetwork that are registered to receive specific multimedia or other broadcasts. See also multicast.

MD5 Message Digest 5. A one-way hashing algorithm that produces a 128-bit hash. Both MD5 and SHA-1 are variations on MD4 and are designed to strengthen the security of the MD4 hashing algorithm. SHA-1 is more secure than MD4 and MD5. Cisco uses hashes for authentication within the IPSec framework. Also used for message authentication in SNMP v.2. MD5 verifies the integrity of the communication, authenticates the origin, and checks for timeliness. MD5 has a smaller digest and is considered to be slightly faster than SHA-1.

MDI Media dependent interface.

MDIX Media dependent interface crossover.

Message Digest A message digest is created by a hash algorithm, such as MD5 or SHA-1, that is used for ensuring message integrity.

MGCP Media Gateway Control Protocol. Media Gateway Control Protocol is a protocol for the control of VoIP calls by external call-control elements known as media gateway controllers or call agents. MGCP merges the IPDC and SGCP protocols.

Mode See Access Modes.

Mode Config See IKE Mode Configuration.

Modular Policy Framework Modular Policy Framework. A means of configuring security appliance features in a manner to similar to Cisco IOS software Modular QoS CLI.

MS mobile station. Refers generically to any mobile device, such as a mobile handset or computer, that is used to access network services. GPRS networks support three classes of MS, which describe the type of operation supported within the GPRS and the GSM mobile wireless networks. For example, a Class A MS supports simultaneous operation of GPRS and GSM services.

MS-CHAP Microsoft CHAP.

MTU Maximum transmission unit, the maximum number of bytes in a packet that can flow efficiently across the network with best response time. For Ethernet, the default MTU is 1500 bytes, but each network can have different values, with serial connections having the smallest values. The MTU is described in RFC 1191.

multicast Multicast refers to a network addressing method in which the source transmits a packet to multiple destinations, a multicast group, simultaneously. See also PIM, SMR.

N

N2H2 A third-party, policy-oriented filtering application that works with the security appliance to control user web access. N2H2 can filter HTTP requests based on destination host name, destination IP address, and username and password. The N2H2 corporation was acquired by Secure Computing in October, 2003.

NAT Network Address Translation. Mechanism for reducing the need for globally unique IP addresses. NAT allows an organization with addresses that are not globally unique to connect to the Internet by translating those addresses into a globally routable address space.

NEM Network Extension Mode. Lets VPN hardware clients present a single, routable network to the remote private network over the VPN tunnel.

NetBIOS Network Basic Input/Output System. A Microsoft protocol that supports Windows host name registration, session management, and data transfer. The security appliance supports NetBIOS by performing NAT of the packets for NBNS UDP port 137 and NBDS UDP port 138.

netmask See mask.

network In the context of security appliance configuration, a network is a group of computing devices that share part of an IP address space and not a single host. A network consists of multiple nodes or hosts. See also host, Internet, intranet, IP, LAN, and node.

NMS network management system. System responsible for managing at least part of a network. An NMS is generally a reasonably powerful and well-equipped computer, such as an engineering workstation. NMSs communicate with agents to help keep track of network statistics and resources.

node Devices such as routers and printers that would not normally be called hosts. See also host, network.

nonvolatile storage, memory Storage or memory that, unlike RAM, retains its contents without power. Data in a nonvolatile storage device survives a power-off, power-on cycle or reboot.

NSAPI Network service access point identifier. One of two components of a GTP tunnel ID, the other component being the IMSI. See also IMSI.

NSSA Not-so-stubby-area. An OSPF feature described by RFC 1587. NSSA was first introduced in Cisco IOS software release 11.2. It is a non-proprietary extension of the existing stub area feature that allows the injection of external routes in a limited fashion into the stub area.

NTLM NT Lan Manager. A Microsoft Windows challenge-response authentication method.

NTP Network time protocol.

O

Oakley A key exchange protocol that defines how to acquire authenticated keying material. The basic mechanism for Oakley is the Diffie-Hellman key exchange algorithm. Oakley is defined in RFC 2412.

object grouping Simplifies access control by letting you apply access control statements to groups of network objects, such as protocol, services, hosts, and networks.

OSPF Open Shortest Path First. OSPF is a routing protocol for IP networks. OSPF is a routing protocol widely deployed in large networks because of its efficient use of network bandwidth and its rapid convergence after changes in topology. The security appliance supports OSPF.

OU Organizational Unit. An X.500 directory attribute.

outbound Refers to traffic whose destination is on an interface with lower security than the source interface.

outbound ACL An ACL applied to outbound traffic.

outside The first interface, usually port 0, that connects to other “untrusted” networks outside the security appliance; the Internet. See also interface, interface names, outbound.

P

PAC PPTP Access Concentrator. A device attached to one or more PSTN or ISDN lines capable of PPP operation and of handling the PPTP protocol. The PAC need only implement TCP/IP to pass traffic to one or more PNSs. It may also tunnel non-IP protocols.

PAT See Dynamic PAT, interface PAT, and Static PAT.

PDP Packet Data Protocol.

Perfmon The security appliance feature that gathers and reports a wide variety of feature statistics, such as connections/second, xlates/second, etc.

PFS Perfect Forwarding Secrecy. PFS enhances security by using different security key for the IPSec Phase 1 and Phase 2 SAs. Without PFS, the same security key is used to establish SAs in both phases. PFS ensures that a given IPSec SA key was not derived from any other secret (like some other keys). In other words, if someone were to break a key, PFS ensures that the attacker would not be able to derive any other key. If PFS were not enabled, someone could hypothetically break the IKE SA secret key, copy all the IPSec protected data, and then use knowledge of the IKE SA secret to compromise the IPSec SA setup by this IKE SA. With PFS, breaking IKE would not give an attacker immediate access to IPSec. The attacker would have to break each IPSec SA individually.

Phase 1 See IPSec Phase 1.

Phase 2 See IPSec Phase 2.

PIM Protocol Independent Multicast. PIM provides a scalable method for determining the best paths for distributing a specific multicast transmission to a group of hosts. Each host has registered using IGMP to receive the transmission. See also PIM-SM.

PIM-SM Protocol Independent Multicast-Sparse Mode. With PIM-SM, which is the default for Cisco routers, when the source of a multicast transmission begins broadcasting, the traffic is forwarded from one MC router to the next, until the packets reach every registered host. See also PIM.

Ping An ICMP request sent by a host to determine if a second host is accessible.

PIX Private Internet eXchange. The Cisco PIX 500-series security appliances range from compact, plug-and-play desktop models for small/home offices to carrier-class gigabit models for the most demanding enterprise and service provider environments. Cisco PIX security appliances provide robust, enterprise-class integrated network security services to create a strong multilayered defense for fast changing network environments.

PKCS12 A standard for the transfer of PKI-related data, such as private keys, certificates, and other data. Devices supporting this standard let administrators maintain a single set of personal identity information.

PNS PPTP Network Server. A PNS is envisioned to operate on general-purpose computing/server platforms. The PNS handles the server side of PPTP. Because PPTP relies completely on TCP/IP and is independent of the interface hardware, the PNS may use any combination of IP interface hardware including LAN and WAN devices.

Policy NAT Lets you identify local traffic for address translation by specifying the source and destination addresses (or ports) in an access list.

POP Post Office Protocol. Protocol that client e-mail applications use to retrieve mail from a mail server.

Pool See IP pool.

Port A field in the packet headers of TCP and UDP protocols that identifies the higher level service which is the source or destination of the packet.

PPP Point-to-Point Protocol. Developed for dial-up ISP access using analog phone lines and modems.

PPTP Point-to-Point Tunneling Protocol. PPTP was introduced by Microsoft to provide secure remote access to Windows networks; however, because it is vulnerable to attack, PPTP is commonly used only when stronger security methods are not available or are not required. PPTP Ports are pptp, 1723/tcp, 1723/udp, and pptp. For more information about PPTP, see RFC 2637. See also PAC, PPTP GRE, PPTP GRE tunnel, PNS, PPTP session, and PPTP TCP.

PPTP GRE Version 1 of GRE for encapsulating PPP traffic.

PPTP GRE tunnel A tunnel defined by a PNS-PAC pair. The tunnel protocol is defined by a modified version of GRE. The tunnel carries PPP datagrams between the PAC and the PNS. Many sessions are multiplexed on a single tunnel. A control connection operating over TCP controls the establishment, release, and maintenance of sessions and of the tunnel itself.

PPTP session PPTP is connection-oriented. The PNS and PAC maintain state for each user that is attached to a PAC. A session is created when end-to-end PPP connection is attempted between a dial user and the PNS. The datagrams related to a session are sent over the tunnel between the PAC and PNS.

PPTP TCP Standard TCP session over which PPTP call control and management information is passed. The control session is logically associated with, but separate from, the sessions being tunneled through a PPTP tunnel.

preshared key A preshared key provides a method of IKE authentication that is suitable for networks with a limited, static number of IPSec peers. This method is limited in scalability because the key must be configured for each pair of IPSec peers. When a new IPSec peer is added to the network, the preshared key must be configured for every IPSec peer with which it communicates. Using certificates and CAs provides a more scalable method of IKE authentication.

primary, primary unit The security appliance normally operating when two units, a primary and secondary, are operating in failover mode.

privileged EXEC mode Privileged EXEC mode lets you to change current settings. Any user EXEC mode command will work in privileged EXEC mode. See also command-specific configuration mode, global configuration mode, user EXEC mode.

protocol, protocol literals A standard that defines the exchange of packets between network nodes for communication. Protocols work together in layers. Protocols are specified in a security appliance configuration as part of defining a security policy by their literal values or port numbers. Possible security appliance protocol literal values are ahp, eigrp, esp, gre, icmp, igmp, igrp, ip, ipinip, ipsec, nos, ospf, pcp, snp, tcp, and udp.

Proxy-ARP Enables the security appliance to reply to an ARP request for IP addresses in the global pool. See also ARP.

public key A public key is one of a pair of keys that are generated by devices involved in public key infrastructure. Data encrypted with a public key can only be decrypted using the associated private key. When a private key is used to produce a digital signature, the receiver can use the public key of the sender to verify that the message was signed by the sender. These characteristics of key pairs provide a scalable and secure method of authentication over an insecure media, such as the Internet.

Q

QoS quality of service. Measure of performance for a transmission system that reflects its transmission quality and service availability.

R

RA Registration Authority. An authorized proxy for a CA. RAs can perform certificate enrollment and can issue CRLs. See also CA, certificate, public key.

RADIUS Remote Authentication Dial-In User Service. RADIUS is a distributed client/server system that secures networks against unauthorized access. RFC 2058 and RFC 2059 define the RADIUS protocol standard. See also AAA and TACACS+.

Refresh Retrieve the running configuration from the security appliance and update the screen. The icon and the button perform the same function.

registration authority See RA.

replay-detection A security service where the receiver can reject old or duplicate packets to defeat replay attacks. Replay attacks rely on the attacker sending out older or duplicate packets to the receiver and the receiver thinking that the bogus traffic is legitimate. Replay-detection is done by using sequence numbers combined with authentication, and is a standard feature of IPSec.

RFC Request for Comments. RFC documents define protocols and standards for communications over the Internet. RFCs are developed and published by IETF.

RIP Routing Information Protocol. Interior gateway protocol (IGP) supplied with UNIX BSD systems. The most common IGP in the Internet. RIP uses hop count as a routing metric.

RLLA Reserved Link Local Address. Multicast addresses range from 224.0.0.0 to 239.255.255.255, however only the range 224.0.1.0 to 239.255.255.255 is available to us. The first part of the multicast address range, 224.0.0.0 to 224.0.0.255, is reserved and referred to as the RLLA. These addresses are unavailable. We can exclude the RLLA range by specifying: 224.0.1.0 to 239.255.255.255. 224.0.0.0 to 239.255.255.255 excluding 224.0.0.0 to 224.0.0.255. This is the same as specifying: 224.0.1.0 to 239.255.255.255.

route, routing The path through a network.

routed firewall mode In routed firewall mode, the security appliance is counted as a router hop in the network. It performs NAT between connected networks and can use OSPF or RIP. See also transparent firewall mode.

RPC Remote Procedure Call. RPCs are procedure calls that are built or specified by clients and executed on servers, with the results returned over the network to the clients.

RSA A public key cryptographic algorithm (named after its inventors, Rivest, Shamir, and Adelman) with a variable key length. The main weakness of RSA is that it is significantly slow to compute compared to popular secret-key algorithms, such as DES. The Cisco implementation of IKE uses a Diffie-Hellman exchange to get the secret keys. This exchange can be authenticated with RSA (or preshared keys). With the Diffie-Hellman exchange, the DES key never crosses the network (not even in encrypted form), which is not the case with the RSA encrypt and sign technique. RSA is not public domain, and must be licensed from RSA Data Security.

RSH Remote Shell. A protocol that allows a user to execute commands on a remote system without having to log in to the system. For example, RSH can be used to remotely examine the status of a number of access servers without connecting to each communication server, executing the command, and then disconnecting from the communication server.

RTCP RTP Control Protocol. Protocol that monitors the QoS of an IPv6 RTP connection and conveys information about the on-going session. See also RTP.

RTP Real-Time Transport Protocol. Commonly used with IP networks. RTP is designed to provide end-to-end network transport functions for applications transmitting real-time data, such as audio, video, or simulation data, over multicast or unicast network services. RTP provides such services as payload type identification, sequence numbering, timestamping, and delivery monitoring to real-time applications.

RTSP Real Time Streaming Protocol. Enables the controlled delivery of real-time data, such as audio and video. RTSP is designed to work with established protocols, such as RTP and HTTP.

rule Conditional statements added to the security appliance configuration to define security policy for a particular situation. See also ACE, ACL, NAT.

running configuration The configuration currently running in RAM on the security appliance. The configuration that determines the operational characteristics of the security appliance.

S

SA security association. An instance of security policy and keying material applied to a data flow. SAs are established in pairs by IPSec peers during both phases of IPSec. SAs specify the encryption algorithms and other security parameters used to create a secure tunnel. Phase 1 SAs (IKE SAs) establish a secure tunnel for negotiating Phase 2 SAs. Phase 2 SAs (IPSec SAs) establish the secure tunnel used for sending user data. Both IKE and IPSec use SAs, although SAs are independent of one another. IPSec SAs are unidirectional and they are unique in each security protocol. A set of SAs are needed for a protected data pipe, one per direction per protocol. For example, if you have a pipe that supports ESP between peers, one ESP SA is required for each direction. SAs are uniquely identified by destination (IPSec endpoint) address, security protocol (AH or ESP), and Security Parameter Index. IKE negotiates and establishes SAs on behalf of IPSec. A user can also establish IPSec SAs manually. An IKE SA is used by IKE only, and unlike the IPSec SA, it is bidirectional.

SCCP Skinny Client Control Protocol. A Cisco-proprietary protocol used between Cisco Call Manager and Cisco VoIP phones.

SCEP Simple Certificate Enrollment Protocol. A method of requesting and receiving (also known as enrolling) certificates from CAs.

SDP Session Definition Protocol. An IETF protocol for the definition of Multimedia Services. SDP messages can be part of SGCP and MGCP messages.

secondary unit The backup security appliance when two are operating in failover mode.

secret key A secret key is a key shared only between the sender and receiver. See key, public key.

security context You can partition a single security appliance into multiple virtual firewalls, known as security contexts. Each context is an independent firewall, with its own security policy, interfaces, and administrators. Multiple contexts are similar to having multiple stand-alone firewalls.

security services See cryptography.

serial transmission A method of data transmission in which the bits of a data character are transmitted sequentially over a single channel.

SGCP Simple Gateway Control Protocol. Controls VoIP gateways by an external call control element (called a call-agent).

SGSN Serving GPRS Support Node. The SGSN ensures mobility management, session management and packet relaying functions.

SHA-1 Secure Hash Algorithm 1. SHA-1 [NIS94c] is a revision to SHA that was published in 1994. SHA is closely modeled after MD4 and produces a 160-bit digest. Because SHA produces a 160-bit digest, it is more resistant to brute-force attacks than 128-bit hashes (such as MD5), but it is slower. Secure Hash Algorithm 1 is a joint creation of the National Institute of Standards and Technology and the National Security Agency. This algorithm, like other hash algorithms, is used to generate a hash value, also known as a message digest, that acts like a CRC used in lower-layer protocols to ensure that message contents are not changed during transmission. SHA-1 is generally considered more secure than MD5.

SIP Session Initiation Protocol. Enables call handling sessions, particularly two-party audio conferences, or “calls.” SIP works with SDP for call signaling. SDP specifies the ports for the media stream. Using SIP, the security appliance can support any SIP VoIP gateways and VoIP proxy servers.

site-to-site VPN A site-to-site VPN is established between two IPSec peers that connect remote networks into a single VPN. In this type of VPN, neither IPSec peer is the destination or source of user traffic. Instead, each IPSec peer provides encryption and authentication services for hosts on the LANs connected to each IPSec peer. The hosts on each LAN send and receive data through the secure tunnel established by the pair of IPSec peers.

SKEME A key exchange protocol that defines how to derive authenticated keying material, with rapid key refreshment.

SMR Stub Multicast Routing. SMR allows the security appliance to function as a “stub router.” A stub router is a device that acts as an IGMP proxy agent. IGMP is used to dynamically register specific hosts in a multicast group on a particular LAN with a multicast router. Multicast routers route multicast data transmissions to hosts that are registered to receive specific multimedia or other broadcasts. A stub router forwards IGMP messages between hosts and MC routers.

SMTP Simple Mail Transfer Protocol. SMTP is an Internet protocol that supports email services.

SNMP Simple Network Management Protocol. A standard method for managing network devices using data structures called Management Information Bases.

split tunneling Allows a remote VPN client simultaneous encrypted access to a private network and clear unencrypted access to the Internet. If you do not enable split tunneling, all traffic between the VPN client and the security appliance is sent through an IPSec tunnel. All traffic originating from the VPN client is sent to the outside interface through a tunnel, and client access to the Internet from its remote site is denied.

spoofing A type of attack designed to foil network security mechanisms such as filters and access lists. A spoofing attack sends a packet that claims to be from an address from which it was not actually sent.

SQL*Net Structured Query Language Protocol. An Oracle protocol used to communicate between client and server processes.

SSH Secure Shell. An application running on top of a reliable transport layer, such as TCP/IP, that provides strong authentication and encryption capabilities.

SSL Secure Sockets Layer. A protocol that resides between the application layer and TCP/IP to provide transparent encryption of data traffic.

standby unit See secondary unit.

stateful inspection Network protocols maintain certain data, called state information, at each end of a network connection between two hosts. State information is necessary to implement the features of a protocol, such as guaranteed packet delivery, data sequencing, flow control, and transaction or session IDs. Some of the protocol state information is sent in each packet while each protocol is being used. For example, a browser connected to a web server uses HTTP and supporting TCP/IP protocols. Each protocol layer maintains state information in the packets it sends and receives. The security appliance and some other firewalls inspect the state information in each packet to verify that it is current and valid for every protocol it contains. This is called stateful inspection and is designed to create a powerful barrier to certain types of computer security threats.

Static PAT Static Port Address Translation. Static PAT is a static address that also maps a local port to a global port. See also Dynamic PAT, NAT.

subnetmask See mask.

T

TACACS+ Terminal Access Controller Access Control System Plus. A client-server protocol that supports AAA services, including command authorization. See also AAA, RADIUS.

TAPI Telephony Application Programming Interface. A programming interface in Microsoft Windows that supports telephony functions.

TCP Transmission Control Protocol. Connection-oriented transport layer protocol that provides reliable full-duplex data transmission.

TCP Intercept With the TCP intercept feature, once the optional embryonic connection limit is reached, and until the embryonic connection count falls below this threshold, every SYN bound for the effected server is intercepted. For each SYN, the security appliance responds on behalf of the server with an empty SYN/ACK segment. The security appliance retains pertinent state information, drops the packet, and waits for the client acknowledgment. If the ACK is received, then a copy of the client SYN segment is sent to the server and the TCP three-way handshake is performed between the security appliance and the server. If this three-way handshake completes, may the connection resume as normal. If the client does not respond during any part of the connection phase, then the security appliance retransmits the necessary segment using exponential back-offs.

TDP Tag Distribution Protocol. TDP is used by tag switching devices to distribute, request, and release tag binding information for multiple network layer protocols in a tag switching network. TDP does not replace routing protocols. Instead, it uses information learned from routing protocols to create tag bindings. TDP is also used to open, monitor, and close TDP sessions and to indicate errors that occur during those sessions. TDP operates over a connection-oriented transport layer protocol with guaranteed sequential delivery (such as TCP). The use of TDP does not preclude the use of other mechanisms to distribute tag binding information, such as piggybacking information on other protocols.

Telnet A terminal emulation protocol for TCP/IP networks such as the Internet. Telnet is a common way to control web servers remotely; however, its security vulnerabilities have led to its replacement by SSH.

TFTP Trivial File Transfer Protocol. TFTP is a simple protocol used to transfer files. It runs on UDP and is explained in depth in RFC 1350.

TID Tunnel Identifier.

TLS Transport Layer Security. A future IETF protocol to replace SSL.

traffic policing The traffic policing feature ensures that no traffic exceeds the maximum rate (bits per second) that you configure, thus ensuring that no one traffic flow can take over the entire resource.

transform set See IPSec transform set.

translate, translation See xlate.

transparent firewall mode A mode in which the security appliance is not a router hop. You can use transparent firewall mode to simplify your network configuration or to make the security appliance invisible to attackers. You can also use transparent firewall mode to allow traffic through that would otherwise be blocked in routed firewall mode. See also routed firewall mode.

transport mode An IPSec encryption mode that encrypts only the data portion (payload) of each packet, but leaves the header untouched. Transport mode is less secure than tunnel mode.

TSP TAPI Service Provider. See also TAPI.

tunnel mode An IPSec encryption mode that encrypts both the header and data portion (payload) of each packet. Tunnel mode is more secure than transport mode.

tunnel A method of transporting data in one protocol by encapsulating it in another protocol. Tunneling is used for reasons of incompatibility, implementation simplification, or security. For example, a tunnel lets a remote VPN client have encrypted access to a private network.

Turbo ACL Increases ACL lookup speeds by compiling them into a set of lookup tables. Packet headers are used to access the tables in a small, fixed number of lookups, independent of the existing number of ACL entries.

U

UDP User Datagram Protocol. A connectionless transport layer protocol in the IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, which requires other protocols to handle error processing and retransmission. UDP is defined in RFC 768.

UMTS Universal Mobile Telecommunication System. An extension of GPRS networks that moves toward an all-IP network by delivering broadband information, including commerce and entertainment services, to mobile users via fixed, wireless, and satellite networks

Unicast RPF Unicast Reverse Path Forwarding. Unicast RPF guards against spoofing by ensuring that packets have a source IP address that matches the correct source interface according to the routing table.

URL Uniform Resource Locator. A standardized addressing scheme for accessing hypertext documents and other services using a browser. For example, http://www.cisco.com.

user EXEC mode User EXEC mode lets you to see the security appliance settings. The user EXEC mode prompt appears as follows when you first access the security appliance. See also command-specific configuration mode, global configuration mode, and privileged EXEC mode.

UTC Coordinated Universal Time. The time zone at zero degrees longitude, previously called Greenwich Mean Time (GMT) and Zulu time. UTC replaced GMT in 1967 as the world time standard. UTC is based on an atomic time scale rather than an astronomical time scale.

UTRAN Universal Terrestrial Radio Access Network. Networking protocol used for implementing wireless networks in UMTS. GTP allows multi-protocol packets to be tunneled through a UMTS/GPRS backbone between a GGSN, an SGSN and the UTRAN.

UUIE User-User Information Element. An element of an H.225 packet that identifies the users implicated in the message.

V

VLAN Virtual LAN. A group of devices on one or more LANs that are configured (using management software) so that they can communicate as if they were attached to the same physical network cable, when in fact they are located on a number of different LAN segments. Because VLANs are based on logical instead of physical connections, they are extremely flexible.

VoIP Voice over IP. VoIP carries normal voice traffic, such as telephone calls and faxes, over an IP-based network. DSP segments the voice signal into frames, which then are coupled in groups of two and stored in voice packets. These voice packets are transported using IP in compliance with ITU-T specification H.323.

VPN Virtual Private Network. A network connection between two peers over the public network that is made private by strict authentication of users and the encryption of all data traffic. You can establish VPNs between clients, such as PCs, or a headend, such as the security appliance.

virtual firewall See security context.

VSA Vendor-specific attribute. An attribute in a RADIUS packet that is defined by a vendor rather than by RADIUS RFCs. The RADIUS protocol uses IANA-assigned vendor numbers to help identify VSAs. This lets different vendors have VSAs of the same number. The combination of a vendor number and a VSA number makes a VSA unique. For example, the cisco-av-pair VSA is attribute 1 in the set of VSAs related to vendor number 9. Each vendor can define up to 256 VSAs. A RADIUS packet contains any VSAs attribute 26, named Vendor-specific. VSAs are sometimes referred to as subattributes.

W

WAN wide-area network. Data communications network that serves users across a broad geographic area and often uses transmission devices provided by common carriers.

WCCP Web Cache Communication Protocol. Transparently redirects selected types of traffic to a group of web cache engines to optimize resource usage and lower response times.

Websense A content filtering solution that manages employee access to the Internet. Websense uses a policy engine and a URL database to control user access to websites.

WEP Wired Equivalent Privacy. A security protocol for wireless LANs, defined in the IEEE 802.11b standard.

WINS Windows Internet Naming Service. A Windows system that determines the IP address associated with a particular network device, also known as “name resolution.” WINS uses a distributed database that is automatically updated with the NetBIOS names of network devices currently available and the IP address assigned to each one.WINS provides a distributed database for registering and querying dynamic NetBIOS names to IP address mapping in a routed network environment. It is the best choice for NetBIOS name resolution in such a routed network because it is designed to solve the problems that occur with name resolution in complex networks.

X

X.509 A widely used standard for defining digital certificates. X.509 is actually an ITU recommendation, which means that it has not yet been officially defined or approved for standardized usage.

xauth See IKE Extended Authentication.

xlate An xlate, also referred to as a translation entry, represents the mapping of one IP address to another, or the mapping of one IP address/port pair to another.



INDEX

Symbols

/bits subnet masks C-3

?

- command string B-4
- help B-4

Numerics

4GE SSM

- connector types 6-2
- fiber 6-3
- SFP 6-3
- support 1-2

802.1Q tagging 5-11

802.1Q trunk 6-7

A

AAA

- about 14-1
- accounting 21-14
- addressing, configuring 33-2
- authentication
  - CLI access 42-5
  - network access 21-1
  - privileged EXEC mode 42-6
- authorization
  - command 42-8
  - downloadable access lists 21-10
  - network access 21-8
- local database support 14-6
- performance 21-1
- server
  - adding 14-9
  - types 14-3
- support summary 14-3
- web clients 21-5

abbreviating commands B-3

Access Control Server 35-2, 35-5, 35-8

access hours, username attribute 32-77

accessing the security appliance using SSL 39-3

accessing the security appliance using TKS1 39-3

access list filter, username attribute 32-79

access lists

- about 18-1
- ACE logging, configuring 18-21
- comments 18-19
- deny flows, managing 18-23
- downloadable 21-10
- EtherType
  - adding 18-10
- EtherType, adding 18-8
- exemptions from posture validation 35-7
- extended
  - about 18-5
  - adding 18-7
- group policy WebVPN filter 32-69
- implicit deny 18-3
- inbound 20-1
- interface, applying 20-2
- IP address guidelines 18-3
- IPsec 29-19
- logging 18-21
- NAT guidelines 18-3
- Network Admission Control, default 35-6

object groups 18-19
outbound 20-1
phone proxy 27-9
remarks 18-19
scheduling activation 18-19
standard, adding 18-11
types 18-2
username for Clientless SSL VPN 32-85
Webtype
adding 18-11
access ports 5-9
ACEs
See access lists
Active/Active failover
about 15-11
actions 15-14
command replication 15-13
configuration synchronization 15-12
configuring
asymmetric routing support 15-37
cable-based failover 15-29
failover criteria 15-36
failover group preemption 15-35
HTTP replication 15-35
interface monitoring 15-36
LAN-based failover 15-31
prerequisites 15-29
virtual MAC addresses 15-36
device initialization 15-12
duplicate MAC addresses, avoiding 15-11, 15-37
primary status 15-12
secondary status 15-12
triggers 15-14
Active/Standby failover
about 15-7
actions 15-10
command replication 15-8
configuration synchronization 15-8
configuring
cable-based 15-21
failover criteria 15-28
HTTP replication 15-26
interface monitoring 15-27
interface poll times 15-41
LAN-based 15-23
prerequisites 15-21
unit poll times 15-41
virtual MAC addresses 15-28
device initialization 15-8
primary unit 15-7
secondary unit 15-7
triggers 15-10
Active Directory, settings for password management 32-27
Active Directory procedures D-14 to ??
Adaptive Security Algorithm 1-17
admin context
about 4-3
changing 7-16
administrative distance 10-3
Advanced Encryption Standard (AES) 29-3
AIP SSM
about 23-1
checking status 23-18
configuration 23-4
loading an image 23-19
sending traffic to 23-8
sessioning to 23-5
support 1-2
alternate address, ICMP message C-15
Application Access Panel, WebVPN 39-57
application access using Clientless SSL VPN
group policy attribute for Clientless SSL VPN 32-70
username attribute for Clientless SSL VPN 32-86
application access using WebVPN
and e-mail proxy 39-79
and hosts file errors 39-44
and Web Access 39-79

configuring client applications 39-78
enabling cookies on browser 39-78
privileges 39-78
quitting properly 39-45
setting up on client 39-78
using e-mail 39-79
with IMAP client 39-79
application inspection
about 26-2
applying 26-5
configuring 26-5
inspection class map 16-12
inspection policy map 16-9
security level requirements 8-1
special actions 16-8
Application Profile Customization Framework 39-54
ARP inspection
about 28-1
enabling 28-2
static entry 28-2
ARP spoofing 28-2
ARP test, failover 15-19
ASA (Adaptive Security Algorithm) 1-17
ASA 5505
Base license 5-2
client
authentication 36-12
configuration restrictions, table 36-2
device pass-through 36-8
group policy attributes pushed to 36-10
mode 36-3
remote management 36-9
split tunneling 36-8
TCP 36-4
trustpoint 36-7
tunnel group 36-7
tunneling 36-5
Xauth 36-4
interfaces, about 5-1
MAC addresses 5-4
maximum VLANs 5-2
native VLAN support 5-11
non-forwarding interface 5-6
power over Ethernet 5-4
protected switch ports 5-9
Security Plus license 5-2
server (headend) 36-1
SPAN 5-4
Spanning Tree Protocol, unsupported 5-9
VLAN interface configuration 5-5
ASDM software
allowing access 42-3
installing 43-2
ASR 15-37
asymmetric routing support 15-37
attributes
RADIUS D-27
username 32-77
attribute-value pairs
TACACS+ D-35
attribute-value pairs (AVP) 32-36
authentication
about 14-2
ASA 5505 as Easy VPN client 36-12
CLI access 42-5
FTP 21-3
HTTP 21-2
network access 21-1
privileged EXEC mode 42-6
restrictions, WebVPN 39-6
Telnet 21-2
web clients 21-5
WebVPN users with digital certificates 39-21
authorization
about 14-2
command 42-8
downloadable access lists 21-10
network access 21-8

Auto-MDI/MDIX 6-2
auto-signon
group policy attribute for Clientless SSL VPN 32-68
username attribute for Clientless SSL VPN 32-87
Auto-Update, configuring 43-19

B

backup device, load balancing 31-6
backup server attributes, group policy 32-53
Baltimore Technologies, CA server support 41-5
banner message, group policy 32-46
basic threat detection
See threat detection
bits subnet masks C-3
Black Ice firewall 32-62
BPDUs
ACL, EtherType 18-10
BPDUs, EtherType access list 18-10
bridge
entry timeout 28-4
table, *See* MAC address table
broadcast Ping test 15-19
bypass authentication 36-8

C

CA
certificate validation, not done in WebVPN 39-2
CRs and 41-2
public key cryptography 41-1
revoked certificates 41-2
server support 41-5
supported servers 41-5
caching 39-52
capturing packets 45-12
cascading access lists 29-15
certificate
authentication, e-mail proxy 39-51
Cisco Unified Mobility 27-56
Cisco Unified Presence 27-61
enrollment protocol 41-7
group matching
configuring 29-9
rule and policy, creating 29-9
Certificate Revocation Lists
See CRLs
certificates
phone proxy 27-17
required by phone proxy 27-18
certification authority
See CA
changing between contexts 7-14
Cisco-AV-Pair LDAP attributes D-12
Cisco Integrated Firewall 32-61
Cisco IP Communicator 27-24
Cisco IP Phones
DHCP 11-4
Cisco IP Phones, application inspection 26-75
Cisco Security Agent 32-61
Cisco Trust Agent 35-8
Cisco UMA. See Cisco Unified Mobility.
Cisco Unified Mobility
architecture 27-53
ASA role 27-2, 27-3
certificate 27-56
functionality 27-52
NAT and PAT requirements 27-54, 27-55
sample configuration 27-75
trust relationship 27-56
Cisco Unified Presence
ASA role 27-2, 27-3
configuring the TLS Proxy 27-62
debugging the TLS Proxy 27-64
NAT and PAT requirements 27-60
sample configuration 27-78
trust relationship 27-61

Cisco UP. See Cisco Unified Presence.
Class A, B, and C addresses C-1
class-default class map 16-5
classes, logging
filtering messages by 44-18
message class variables 44-18, E-5
types 44-18, E-5
classes, MPF
See class map
classes, resource
See resource management
class map
inspection 16-12
Layer 3/4
management traffic 16-7
match commands 16-5
through traffic 16-5
regular expression 16-16
CLI
abbreviating commands B-3
adding comments B-7
command line editing B-3
command output paging B-6
displaying B-6
help B-4
paging B-6
syntax formatting B-3
client
VPN 3002 hardware, forcing client update 31-4
Windows, client update notification 31-4
client access rules, group policy 32-63
client firewall, group policy 32-60
clientless authentication 35-8
Clientless SSL VPN
configuring for specific users 32-81
client mode 36-3
client update, performing 31-4
cluster
IP address, load balancing 31-6
load balancing configurations 31-7
mixed scenarios 31-8
virtual 31-6
command authorization
about 42-9
configuring 42-8
multiple contexts 42-10
command prompts B-2
comments
access lists 18-19
configuration B-7
configuration
clearing 2-9
comments B-7
factory default
commands 2-1
restoring 2-2
saving 2-6
text file 2-9
URL for a context 7-9
viewing 2-8
configuration mode
accessing 2-5
prompt B-2
connection blocking 24-22
connection limits
configuring 24-17
per context 7-6
connect time, maximum, username attribute 32-79
console port logging 44-10
content transformation, WebVPN 39-52
contexts
See security contexts
conversion error, ICMP message C-16
cookies, enabling for WebVPN 39-7
CRACK protocol 29-27
crash dump 45-13
crypto map
acccess lists 29-19

applying to interfaces 29-19, 38-7
clearing configurations 29-27
creating an entry to use the dynamic crypto map 34-7
definition 29-12
dynamic 29-24
dynamic, creating 34-6
entries 29-12
examples 29-20
policy 29-13
crypto show commands 29-26
CSC SSM
about 23-10
checking status 23-18
failover 23-11
getting started 23-12
loading an image 23-19
sending traffic to 23-16
support 1-2
what to scan 23-13
custom firewall 32-62
customization, Clientless SSL VPN
group policy attribute 32-66
login windows for users 32-27
username attribute 32-83
username attribute for Clientless SSL VPN 32-23
cut-through proxy 21-1

D

data flow
routed firewall 17-1
transparent firewall 17-11
DDNS 11-6
debugging IPSec 30-9
debug messages 45-12
default
class 7-3
DefaultL2Lgroup 32-1
DefaultRAgroup 32-1
domain name, group policy 32-49
group policy 32-1, 32-36
LAN-to-LAN tunnel group 32-16
remote access tunnel group, configuring 32-7
routes, defining equal cost routes 10-4
tunnel group 29-11, 32-2
default configuration
commands 2-1
restoring 2-2
default policy 16-3
default routes
about 10-4
configuring 10-4
deleting files from Flash 43-2
deny flows, logging 18-23
deny in a crypto map 29-15
deny-message
group policy attribute for Clientless SSL VPN 32-66
username attribute for Clientless SSL VPN 32-84
DES, IKE policy keywords (table) 29-3
device ID, including in messages 44-20
device pass-through, ASA 5505 as Easy VPN client 36-8
DfltGrpPolicy 32-37
DHCP
addressing, configuring 33-3
Cisco IP Phones 11-4
options 11-3
relay 11-5
server 11-1, 11-2
transparent firewall 18-6
DHCP Intercept, configuring 32-50
Diffie-Hellman
Group 5 29-4
groups supported 29-4
DiffServ preservation 25-5
digital certificates
authenticating WebVPN users 39-21
SSL 39-6
WebVPN authentication restrictions 39-6

directory hierarchy search D-4
disabling content rewrite 39-53
disabling messages, specific message IDs 44-22
DMZ, definition 1-14
DNS
  dynamic 11-6
  inspection
    about 26-14
    managing 26-14
    rewrite, about 26-15
    rewrite, configuring 26-16
  NAT effect on 19-17
  server, configuring 32-40
domain attributes, group policy 32-48
domain name 9-2
dotted decimal subnet masks C-3
downloadable access lists
  configuring 21-10
  converting netmask expressions 21-14
DSCP preservation 25-5
DUAL 10-25
dual IP stack, configuring 13-4
dual-ISP support 10-5
duplex, configuring 6-2
dynamic crypto map 29-24
  creating 34-6
  See also crypto map
Dynamic DNS 11-6
dynamic NAT
  See NAT

E

Easy VPN
  client
    authentication 36-12
    configuration restrictions, table 36-2
    enabling and disabling 36-1
    group policy attributes pushed to 36-10
    mode 36-3
    remote management 36-9
    trustpoint 36-7
    tunnels 36-9
    Xauth 36-4
  server (headend) 36-1
Easy VPN client
  ASA 5505
    device pass-through 36-8
    split tunneling 36-8
    TCP 36-4
    tunnel group 36-7
    tunneling 36-5
echo reply, ICMP message C-15
ECMP 10-3
editing command lines B-3
egress VLAN for VPN sessions 32-43
EIGRP 18-6
  configuring 10-26
  DUAL algorithm 10-25
  hello interval 10-30
  hello packets 10-25
  hold time 10-25, 10-30
  neighbor discovery 10-25
  Overview 10-25
  stub routing 10-27
  stuck-in-active 10-25
e-mail
  configuring for WebVPN 39-50
  proxies, WebVPN 39-50
  proxy, certificate authentication 39-51
  WebVPN, configuring 39-50
EMBLEM format, using in logs 44-21
enable command 2-5
end-user interface, WebVPN, defining 39-56
Enterprises 11-4
Entrust, CA server support 41-5
established command, security level requirements 8-2
Ethernet

Auto-MDI/MDIX 6-2
duplex 6-2
speed 6-2
EtherType
assigned numbers 18-10
See also access lists
external group policy, configuring 32-38

F

facility, syslog 44-9
factory default configuration
commands 2-1
restoring 2-2
failover
about 15-1
Active/Active, configuring 15-29
Active/Active, *See* Active/Active failover
Active/Standby, configuring 15-21
Active/Standby, *See* Active/Standby failover
configuration file
terminal messages, Active/Active 15-12
terminal messages, Active/Standby 15-8
configuring 15-20
contexts 15-7
controlling 15-51
debug messages 15-53
disabling 15-52
displaying commands 15-50
encrypting failover communication 15-41
Ethernet failover cable 15-4
examples
Active/Active LAN-based failover A-25, A-30
Active/Standby cable-based failover A-34, A-35
Active/Standby LAN-based failover A-24, A-28
failover link 15-3
forcing 15-51
health monitoring 15-18
interface health 15-18
interface monitoring 15-18
interface tests 15-18
licenses 15-3
link communications 15-3
MAC addresses
about 15-7
automatically assigning 7-12
monitoring, configuration 15-51
monitoring, health 15-18
network tests 15-19
primary unit 15-7
redundant interfaces 6-5
restoring a failed group 15-52
restoring a failed unit 15-52
secondary unit 15-7
serial cable 15-4
SNMP syslog traps 15-53
software versions 15-3
Stateful Failover, *See* Stateful Failover
state link 15-5
subsecond 15-41
system log messages 15-53
system requirements 15-2
testing 15-51
type selection 15-15
understanding 15-1
unit health 15-18
verifying the configuration 15-42
fast path 1-18
fiber interfaces 6-3
filter (access list)
group policy attribute for Clientless SSL VPN 32-69
username attribute for Clientless SSL VPN 32-85
filtering
about 22-1
ActiveX 22-2
FTP 22-9
Java applets 22-3
security level requirements 8-2

servers supported 22-4
show command output B-4
URLs 22-4
firewall
Black Ice 32-62
Cisco Integrated 32-61
Cisco Security Agent 32-61
custom 32-62
Network Ice 32-62
none 32-62
Sygate personal 32-62
Zone Labs 32-62
firewall mode
about 17-1
configuring 2-5
firewall policy, group policy 32-60
Flash memory
removing files 43-2
FO (failover) license 15-3
FO_AA license 15-3
format of messages 44-24
fragmentation policy, IPsec 29-7
fragment protection 1-15
fragment size 24-22
FTP inspection
about 26-28
configuring 26-27

G

general attributes, tunnel group 32-3
general parameters, tunnel group 32-3
general tunnel-group connection parameters 32-3
generating RSA keys 41-6
global addresses
recommendations 19-16
specifying 19-26
global e-mail proxy attributes 39-50
global IPsec SA lifetimes, changing 29-21
group-lock, username attribute 32-80
group policy
address pools 32-60
attributes 32-40
backup server attributes 32-53
client access rules 32-63
configuring 32-38
default domain name for tunneled packets 32-49
definition 32-1, 32-36
domain attributes 32-48
Easy VPN client, attributes pushed to ASA 5505 36-10
external, configuring 32-38
firewall policy 32-60
hardware client user idle timeout 32-51
internal, configuring 32-39
IP phone bypass 32-52
IPSec over UDP attributes 32-46
LEAP Bypass 32-52
network extension mode 32-53
security attributes 32-44
split tunneling attributes 32-47
split-tunneling domains 32-49
user authentication 32-51
VPN attributes 32-41
VPN hardware client attributes 32-50
webvpn attributes 32-65
WINS and DNS servers 32-40
group policy, default 32-36
group policy, secure unit authentication 32-50
group policy attributes for Clientless SSL VPN
application access 32-70
auto-signon 32-68
customization 32-66
deny-message 32-66
filter 32-69
home page 32-68
html-content filter 32-67
keep-alive-ignore 32-71

port forward 32-70
port-forward-name 32-71
sso-server 32-72
svc 32-73
url-list 32-69
GTP inspection
about 26-33
configuring 26-33

H

H.225 timeouts 26-44
H.245 troubleshooting 26-45
H.323
transparent firewall guidelines 17-8
H.323 inspection
about 26-39
configuring 26-39
limitations 26-41
troubleshooting 26-45
hairpinning 29-19
hardware client, group policy attributes 32-50
help, command line B-4
HMAC hashing method 29-3
hold-period 35-11
homepage
group policy attribute for Clientless SSL VPN 32-68
username attribute for Clientless SSL VPN 32-83
hostname
configuring 9-2
in banners 9-2
multiple context mode 9-2
hosts, subnet masks for C-3
hosts file
errors 39-44
reconfiguring 39-46
WebVPN 39-45
HSRP 17-8
html-content-filter
group policy attribute for Clientless SSL VPN 32-67
username attribute for Clientless SSL VPN 32-82
HTTP(S)
authentication 42-6
filtering 22-4
HTTP/HTTPS Web VPN proxy, setting 39-7
HTTP compression, Clientless SSL VPN, enabling 32-72, 32-88
HTTP inspection
about 26-46
configuring 26-46
HTTP redirection for login, Easy VPN client on the ASA 5505 36-12
HTTPS for WebVPN sessions 39-3, 39-4
hub-and-spoke VPN scenario 29-19

I

ICMP
testing connectivity 45-1
type numbers C-15
idle timeout
hardware client user, group policy 32-51
username attribute 32-78
ID method for ISAKMP peers, determining 29-6
IKE
benefits 29-2
creating policies 29-4
keepalive setting, tunnel group 32-4
pre-shared key, Easy VPN client on the ASA 5505 36-7
See also ISAKMP
ILS inspection 26-54
IM 26-69
inbound access lists 20-1
Individual user authentication 36-12
information reply, ICMP message C-15
information request, ICMP message C-15
inheritance

tunnel group 32-1
username attribute 32-77
inside, definition 1-14
inspection_default class-map 16-4
inspection engines
See application inspection
Instant Messaging inspection 26-69
intercept DHCP, configuring 32-50
interfaces
ASA 5505
about 5-1
enabled status 5-9
IP address 5-7
MAC addresses 5-4
maximum VLANs 5-2
non-forwarding 5-6
protected switch ports 5-9
switch port configuration 5-9
trunk ports 5-11
VLAN interface configuration 5-5
configuring for remote access 34-2
configuring IPv6 on 13-3
duplex 6-2
enabled status 6-2
enabling 6-3
failover monitoring 15-18
fiber 6-3
global addresses 19-26
IDs 6-2, 8-4
IP address 8-5
MAC addresses
automatically assigning 7-11
manually assigning to interfaces 8-5
mapped name 7-8
naming, physical and subinterface 8-4
naming, VLAN 5-6
redundant 6-4
SFP 6-3
speed 6-2
subinterfaces 6-7
viewing monitored interface status 15-50
internal group policy, configuring 32-39
Internet Security Association and Key Management Protocol
See ISAKMP
intrusion prevention configuration 23-4
IP addresses
ASA 5505 5-7
classes C-1
configuring an assignment method for remote access clients 33-1
configuring for VPNs 33-1
configuring local IP address pools 33-2
interface 8-5
management, transparent firewall 9-5
private C-2
subnet mask C-4
IP phone 36-8
phone proxy provisioning 27-13
IP phone bypass, group policy 32-52
IP phones
addressing requirements for phone proxy 27-11
supported for phone proxy 27-12
IPS configuration 23-4
IPSec
anti-replay window 25-12
enabling debug 30-9
modes 30-2
over UDP, group policy, configuring attributes 32-46
remote-access tunnel group 32-8
setting maximum active VPN sessions 31-3
IPsec
access list 29-19
basic configuration with static crypto maps 29-22
Cisco VPN Client 29-2
configuring 29-1, 29-11
crypto map entries 29-12
fragmentation policy 29-7

over NAT-T, enabling 29-7
over TCP, enabling 29-8
SA lifetimes, changing 29-21
tunnel 29-11
viewing configuration 29-26
IPSec parameters, tunnel group 32-4
ipsec-ra, creating an IPSec remote-access tunnel 32-8
IP spoofing, preventing 24-21
IPv6
access lists 13-6
commands 13-1
configuring alongside IPv4 13-4
default route 13-5
dual IP stack 13-4
duplicate address detection 13-4
enabling 13-3
neighbor discovery 13-7
router advertisement messages 13-9
static neighbor 13-11
static routes 13-5
verifying 13-11
IPv6 addresses
anycast C-9
command support for 13-1
format C-5
multicast C-8
prefixes C-10
required C-10
types of C-6
unicast C-6
IPv6 VPN
access, enabling with CLI 32-12
ISAKMP
about 29-2
configuring 29-1, 29-2
determining an ID method for peers 29-6
disabling in aggressive mode 29-6
enabling on the outside interface 29-6, 34-3
keepalive setting, tunnel group 32-4
policies, configuring 29-5
See also IKE

J

Java applets, filtering 22-2
Java object signing 39-53
java-trustpoint 39-53
jumbo frames
Ethernet
jumbo frames 5-1, 6-1

K

keep-alive-ignore
group policy attribute for Clientless SSL VPN 32-71
username attribute for Clientless SSL VPN 32-87
Kerberos
configuring 14-9
support 14-6

L

L2TP description 30-1
LAN-to-LAN tunnel group, configuring 32-16
latency
about 25-1
configuring 25-2, 25-3
reducing 25-7
Layer 2 firewall
See transparent firewall
Layer 2 forwarding table
See MAC address table
Layer 2 Tunneling Protocol 30-1
Layer 3/4
matching multiple policy maps 16-21
LCS Federation Scenario 27-60
LDAP

AAA support 14-12
application inspection 26-54
attribute mapping 14-15
Cisco-AV-pair D-12
configuring 14-9
configuring a AAA server D-3 to ??
directory search D-4
example configuration procedures D-14 to ??
hierarchy example D-4
SASL 14-13
server type 14-13
user authentication 14-13
user authorization 14-14
LEAP Bypass, group policy 32-52
licenses
Cisco Unified Communications Proxy features 27-4
FO 15-3
FO_AA 15-3
managing 3-1
UR 15-3
link up/down test 15-18
LLQ
See low-latency queue
load balancing
cluster configurations 31-7
concepts 31-6
eligible clients 31-7
eligible platforms 31-7
implementing 31-6
mixed cluster scenarios 31-8
platforms 31-7
prerequisites 31-7
local user database
adding a user 14-8
configuring 14-7
logging in 42-7
support 14-6
lockout recovery 42-19
log buffer
save to internal Flash 44-15
send to FTP server 44-15
logging
access lists 18-21
classes
filtering messages by 44-17
types 44-18, E-5
device-id, including in system log messages 44-20
e-mail
configuring as output destination 44-10
destination address 44-11
source address 44-11
EMBLEM format 44-21
facility option 44-9
filtering
by message class 44-18
by message list 44-18
by severity level 44-6
logging queue, configuring 44-20
output destinations
ASDM 44-11
console port 44-10
email address 44-10
internal buffer 44-6
SNMP 44-5
syslog serversyslog server
configuring as output destination 44-9
Telnet or SSH session 44-6
queue
changing the size of 44-20
configuring 44-20
viewing queue statistics 44-20
severity level
changing 44-22
severity level, changing 44-22
timestamp, including 44-20
login
banner, configuring 42-20
console 2-5

enable 2-5
FTP 21-3
global configuration mode 2-5
local user 42-7
password 9-1
simultaneous, username attribute 32-78
SSH 42-3
Telnet 9-1
windows, customizing for users of Clientless SSL VPN sessions 32-27
low-latency queue
applying 25-2, 25-3

M

MAC address
redundant interfaces 6-5
MAC addresses
ASA 5505 5-4
ASA 5505 device pass-through 36-8
automatically assigning 7-11
failover 15-7
manually assigning to interfaces 8-5
security context classification 4-4
MAC address table
about 17-11
built-in-switch 28-3
entry timeout 28-4
MAC learning, disabling 28-4
resource management 7-6
static entry 28-3
MAC learning, disabling 28-4
management IP address, transparent firewall 9-5
man-in-the-middle attack 28-2
mapped interface name 7-8
mask
reply, ICMP message C-16
request, ICMP message C-15
match commands
inspection class map 16-10
Layer 3/4 class map 16-5
matching, certificate group 29-9
maximum active IPSec VPN sessions, setting 31-3
maximum connect time,username attribute 32-79
maximum object size to ignore username attribute for Clientless SSL VPN 32-87
maximum sessions, IPSec 31-13
MD5, IKE policy keywords (table) 29-3
media termination address, criteria 27-9
message list
filtering by 44-18
message-of-the-day banner 42-20
messages, logging
classes
about 44-17
list of 44-18, E-5
component descriptions 44-24
filtering by message list 44-18
format of 44-24
message list, creating 44-18
severity levels 44-25
metacharacters, regular expression 16-13, B-5
MGCP inspection
about 26-55
configuring 26-55
MIBs 44-1
Microsoft Access Proxy 27-59
Microsoft Active Directory, settings for password management 32-27
Microsoft Internet Explorer client parameters, configuring 32-54
Microsoft Windows 2000 CA, supported 41-5
mixed cluster scenarios, load balancing 31-8
mixed-mode Cisco UCM cluster, configuring for phone proxy 27-19
MMP inspection 27-52
mobile redirection, ICMP message C-16
mode
context 4-10

firewall 2-5
Modular Policy Framework
See MPF
monitoring
failover 15-18
OSPF 10-19
resource management 7-19
SNMP 44-1
monitoring devices with CS-MARS E-3
monitoring switch traffic, ASA 5505 5-4
More prompt B-6
MPF
about 16-1
default policy 16-3
examples 16-24
feature directionality 16-18
features 16-1
flows 16-21
matching multiple policy maps 16-21
service policy, applying 16-23
See also class map
See also policy map
MPLS
LDP 18-10
router-id 18-10
TDP 18-10
MSIE client parameters, configuring 32-54
MTU size, Easy VPN client, ASA 5505 36-5
multicast traffic 17-8
multiple context mode
See security contexts

N

NAC
See Network Admission Control
naming an interface
ASA 5505 5-6
other models 8-4
NAT
about 19-1
bypassing NAT
about 19-11
configuration 19-32
DNS 19-17
dynamic NAT
about 19-6
configuring 19-25
implementation 19-19
examples 19-36
exemption from NAT
about 19-11
configuration 19-35
identity NAT
about 19-11
configuration 19-32
NAT ID 19-19
order of statements 19-16
overlapping addresses 19-36
PAT
about 19-8
configuring 19-25
implementation 19-19
policy NAT
about 19-11
port redirection 19-38
RPC not supported with 26-81
same security level 19-15
security level requirements 8-2
static identify, configuring 19-33
static NAT
about 19-9
configuring 19-28
static PAT
about 19-9
configuring 19-29
transparent mode 19-3
types 19-6

native VLAN support 5-11
NAT-T
enabling IPsec over NAT-T 29-7
using 29-7
Netscape CMS, CA server support 41-5
Network Activity test 15-19
Network Admission Control
Access Control Server 35-5
ACL, default 35-6
clientless authentication 35-8
configuring 32-56
exemptions 35-7
port 35-10
retransmission retries 35-11
retransmission retry timer 35-10
revalidation timer 35-6
session reinitialization timer 35-11
uses, requirements, and limitations 35-1
network extension mode 36-3
network extension mode, group policy 32-53
Network Ice firewall 32-62
networks, overlapping 19-36
Nokia VPN Client 29-27
non-secure Cisco UCM cluster, configuring phone proxy 27-13
NTLM support 14-6
NT server
configuring 14-9
support 14-6

O

object groups
nesting 18-16
removing 18-19
open ports C-14
operating systems, posture validation exemptions 35-7
OSPF
about 10-9
area authentication 10-14
area MD5 authentication 10-14
area parameters 10-14
authentication key 10-12
cost 10-12
dead interval 10-12
default route 10-17
displaying update packet pacing 10-19
enabling 10-10
hello interval 10-12
interface parameters 10-12
link-state advertisement 10-9
logging neighbor states 10-18
MD5 authentication 10-12
monitoring 10-19
NSSA 10-15
packet pacing 10-19
processes 10-9
redistributing routes 10-10
route calculation timers 10-18
route map 10-7
route summarization 10-16
stub area 10-14
summary route cost 10-14
outbound access lists 20-1
Outlook Web Access (OWA) and WebVPN 39-79
output destinations 44-6
e-mail address 44-6, 44-10
SNMP management station 44-6
specifying 44-10
syslog server 44-6, 44-8
Telnet or SSH session 44-6
viewing logs 44-8
outside, definition 1-14
oversubscribing resources 7-2

P

packet

capture 45-12
classifier 4-3
packet flow
routed firewall 17-1
transparent firewall 17-11
paging screen displays B-6
parameter problem, ICMP message C-15
password
resetting on SSM hardware module 45-10
password management, Active Directory settings 32-27
passwords
changing 9-1
clientless authentication 35-9
recovery 45-6
security appliance 9-1
username, setting 32-76
WebVPN 39-74
password-storage, username attribute 32-81
PAT
Easy VPN client mode 36-3
See also NAT
static 19-29
PDA support for WebVPN 39-50
peers
alerting before disconnecting 29-9
ISAKMP, determining ID method 29-6
performance, optimizing for WebVPN 39-52
permit in a crypto map 29-15
phone proxy
access lists 27-9
ASA role 27-3
certificates 27-17
Cisco IP Communicator 27-24
Cisco UCM supported versions 27-12
configuration prerequisites 27-9
configuring mixed-mode Cisco UCM cluster 27-19
configuring non-secure Cisco UCM cluster 27-13
event recovery 27-41
IP phone addressing 27-11
IP phone provisioning 27-13
IP phones supported 27-12
Linksys routers, configuring 27-24
NAT and PAT requirements 27-10
ports 27-10
rate limiting 27-25
required certificates 27-18
sample configurations 27-65
SAST keys 27-41
TLS Proxy on ASA, described 27-3
troubleshooting 27-26
ping
See ICMP
PKI protocol 41-7
PoE 5-4
policing
flow within a tunnel 25-9
policy, QoS 25-1
policy map
inspection 16-9
Layer 3/4
about 16-17
adding 16-22
default policy 16-21
feature directionality 16-18
flows 16-21
policy NAT
about 19-11
dynamic, configuring 19-25
static, configuring 19-28
static PAT, configuring 19-30
pools, address
DHCP 11-2
global NAT 19-26
port-forward
group policy attribute for Clientless SSL VPN 32-70
username attribute for Clientless SSL VPN 32-86
port forwarding
configuring client applications 39-78

port-forward-name
group policy attribute for Clientless SSL VPN 32-71
username attribute for Clientless SSL VPN 32-87
ports
open on device C-14
phone proxy 27-10
redirection, NAT 19-38
TCP and UDP C-11
posture validation
exemptions 35-7
port 35-10
revalidation timer 35-6
uses, requirements, and limitations 35-1
power over Ethernet 5-4
PPPoE, configuring 37-1 to 37-5
pre-shared key, Easy VPN client on the ASA 5505 36-7
primary unit, failover 15-7
printers 36-8
private networks C-2
privileged EXEC mode, accessing 2-5
privileged mode
accessing 2-5
prompt B-2
privilege level, username, setting 32-76
prompts
command B-2
more B-6
protocol numbers and literal values C-11
proxy
See e-mail proxy
proxy bypass 39-53
proxy servers
SIP and 26-69
public key cryptography 41-1

Q

QoS
about 25-1, 25-3
DiffServ preservation 25-5
DSCP preservation 25-5
feature interaction 25-4
policies 25-1
priority queueing
IPSec anti-replay window 25-12
statistics 25-13
token bucket 25-2
traffic shaping
overview 25-4
viewing statistics 25-13
Quality of Service
See QoS
question mark
command string B-4
help B-4
queue, logging
changing the size of 44-20
viewing statistics 44-20
queue, QoS
latency, reducing 25-7
limit 25-2, 25-3

R

RADIUS
attributes D-27
Cisco AV pair D-12
configuring a AAA server D-27
configuring a server 14-9
downloadable access lists 21-10
network access authentication 21-3
network access authorization 21-10
support 14-4
RAS, H.323 troubleshooting 26-45
rate limiting 25-3
rate limiting, phone proxy 27-25
RealPlayer 26-65
reboot, waiting until active sessions end 29-8

redirect, ICMP message C-15
redundancy, in site-to-site VPNs, using crypto maps 29-26
redundant interfaces
  configuring 6-6
  failover 6-5
  MAC address 6-5
  setting the active interface 6-7
Registration Authority description 41-2
regular expression 16-13
reloading
  context 7-17
  security appliance 45-6
remarks 18-19
remote access
  configuration summary 34-1
  IPSec tunnel group, configuring 32-8
  restricting 32-80
  tunnel group, configuring default 32-7
  user, adding 34-4
  VPN, configuring 34-1
remote management, ASA 5505 36-9
resetting the SSM hardware module password 45-10
resource management
  about 7-2
  assigning a context 7-10
  class 7-4
  configuring 7-1
  default class 7-3
  monitoring 7-19
  oversubscribing 7-2
  resource types 7-6
  unlimited 7-2
resource usage 7-22
retransmission retries, Network Admission Control 35-11
retransmission retry timer, Network Admission Control 35-10
revalidation timer, Network Admission Control 35-6
revoked certificates 41-2
rewrite, disabling 39-53
RIP
  about 10-20
  enabling 10-21
routed mode
  about 17-1
  setting 2-5
route maps
  defining 10-7
  uses 10-7
router
  advertisement, ICMP message C-15
  solicitation, ICMP message C-15
routes
  about default 10-4
  about static 10-2
  configuring default routes 10-4
  configuring IPv6 default 13-5
  configuring IPv6 static 13-5
  configuring static routes 10-3
routing
  OSPF 10-20
  other protocols 18-6
RS-232 cable
  See failover 15-4
RSA
  KEON, CA server support 41-5
  keys, generating 41-6, 42-2
  signatures, IKE authentication method 41-2
RTSP inspection
  about 26-65
  configuring 26-65
running configuration
  copying 43-8
  saving 2-6

S

same security level communication

enabling 8-7
NAT 19-15
SAs, lifetimes 29-21
SAST keys 27-41
SCCP (Skinny) inspection
about 26-75
configuration 26-75
configuring 26-75
SDI
configuring 14-9
support 14-5
secondary device, virtual cluster 31-6
secondary unit, failover 15-7
secure unit authentication 36-12
secure unit authentication, group policy 32-50
security, WebVPN 39-2, 39-8
Security Agent, Cisco 32-61
security appliance
CLI B-1
connecting to 2-4
CS-MARS interoperability E-1
managing licenses 3-1
managing the configuration 2-6
reloading 45-6
upgrading software 43-2
viewing files in Flash memory 43-1
security association
clearing 29-26
See also SAs
security attributes, group policy 32-44
security contexts
about 4-1
adding 7-7
admin context
about 4-3
changing 7-16
assigning to a resource class 7-10
cascading 4-8
changing between 7-14
classifier 4-3
command authorization 42-10
configuration
URL, changing 7-16
URL, setting 7-9
logging in 4-9
MAC addresses
automatically assigning 7-11
classifying using 4-4
managing 7-1, 7-15
mapped interface name 7-8
monitoring 7-18
multiple mode, enabling 4-10
nesting or cascading 4-9
prompt B-2
reloading 7-17
removing 7-15
resource management 7-2
resource usage 7-22
saving all configurations 2-7
unsupported features 4-2
VLAN allocation 7-7
security level
about 8-1
interface 8-4
interface, ASA 5505 5-6
serial cable
See failover
server group 35-5
service policy
applying 16-23
default 16-24
global 16-24
interface 16-24
session management path 1-17
session reinitialization timer, Network Admission Control 35-11
severity levels, of system log messages
changing 44-6

filtering by 44-6
list of 44-25
severity levels, of system messages
definition 44-25
SHA, IKE policy keywords (table) 29-3
show command, filtering output B-4
simultaneous logins, username attribute 32-78
single mode
backing up configuration 4-10
configuration 4-10
enabling 4-10
restoring 4-11
single sign-on
See SSO
single-signon
group policy attribute for Clientless SSL VPN 32-72
username attribute for Clientless SSL VPN 32-88
SIP inspection
about 26-69
configuring 26-68
instant messaging 26-69
timeouts 26-74
troubleshooting 26-74
site-to-site VPNs, redundancy 29-26
smart tunnels 39-30
SMTP inspection 26-78
SNMP
about 44-1
management station 44-6
MIBs 44-1
traps 44-2
source quench, ICMP message C-15
SPAN 5-4
Spanning Tree Protocol, unsupported 5-9
speed, configuring 6-2
split tunneling
ASA 5505 as Easy VPN client 36-8
group policy 32-47
group policy, domains 32-49
SSH
authentication 42-6
concurrent connections 42-2
login 42-3
password 9-1
RSA key 42-2
username 42-3
SSL
certificate 39-6
used to access the security appliance 39-3
SSL/TLS encryption protocols
configuring 39-6
WebVPN 39-6
SSL VPN Client
compression 40-14
DPD 40-12
enabling 40-3
address assignment 40-3
permanent installation 40-5
tunnel group 40-4
group policy attribute for Clientless SSL VPN 32-73
installing 40-2
images 40-2
order 40-2
keepalive messages 40-13
logging out sessions 40-15
username attribute for Clientless SSL VPN 32-89
viewing sessions 40-15
SSM
checking status 23-18
configuration
AIP SSM 23-4
CSC SSM 23-12
loading an image 23-19
See also AIP SSM
See also CSC SSM
sso-server
group policy attribute for Clientless SSL VPN 32-72
username attribute for Clientless SSL VPN 32-88

SSO with WebVPN 39-8 to 39-20
configuring HTTP Basic and NTLM authentication 39-9
configuring HTTP form protocol 39-15
configuring SiteMinder 39-10, 39-12
startup configuration
copying 43-8
saving 2-6
Stateful Failover
about 15-16
state information 15-17
state link 15-5
statistics 15-45, 15-49
stateful inspection 1-17
state information 15-17
state link 15-5
static ARP entry 28-2
static bridge entry 28-3
static NAT
See NAT
static PAT
See PAT
static routes
about 10-2
configuring 10-3
tracking 10-5
statistics, QoS 25-13
stealth firewall
See transparent firewall
stuck-in-active 10-25
subcommand mode prompt B-2
subinterfaces, adding 6-7
subnet masks
/bits C-3
about C-2
address range C-4
determining C-3
dotted decimal C-3
number of hosts C-3
Sun Microsystems Java™ Runtime Environment (JRE) and WebVPN 39-40
Sun Microsystems Java Runtime Environment and WebVPN 39-78
Sun RPC inspection
about 26-81
configuring 26-81
SVC
See SSL VPN Client
svc
group policy attribute for Clientless SSL VPN 32-73
username attribute for Clientless SSL VPN 32-89
switch MAC address table 28-3
switch ports
access ports 5-9
default configuration 5-4
protected 5-9
SPAN 5-4
trunk ports 5-11
Sygate Personal Firewall 32-62
SYN attacks, monitoring 7-23
SYN cookies 7-23
syntax formatting B-3
syslog server
as output destination 44-8
designating 44-9
designating more than one 44-9
EMBLEM format
configuring 44-21
enabling 44-9
system configuration 4-3
system log messages
classes 44-18, E-5
classes of 44-17
configuring in groups
by message list 44-18
by severity level 44-6
creating lists of 44-17
device ID, including 44-20

disabling logging of 44-6
filtering by message class 44-17
managing in groups
by message class 44-18
creating a message list 44-17
output destinations 44-6
email address 44-10
SNMP 44-5
syslog message server 44-6
Telnet or SSH session 44-6
severity levels
about 44-25
changing the severity level of a message 44-6
timestamp, including 44-20

T

TACACS+
command authorization, configuring 42-14
configuring a server 14-9
network access authorization 21-8
support 14-5
tail drop 25-3
TCP
ASA 5505 as Easy VPN client 36-4
connection limits per context 7-6
ports and literal values C-11
sequence number randomization
disabling in NAT configuration 19-26
disabling using Modular Policy Framework 24-19
TCP Intercept
enabling using Modular Policy Framework 24-19
enabling using NAT 19-26
monitoring 7-23
TCP normalization 24-12
Telnet
allowing management access 42-1
authentication 42-6
concurrent connections 42-1
password 9-1
testing configuration 45-1
threat detection
basic
drop types 24-2
enabling 24-2
overview 24-2
rate intervals 24-2
rate intervals, setting 24-3
statistics, clearing 24-4
statistics, viewing 24-4
system performance 24-2
scanning
attackers, viewing 24-7
default limits, changing 24-6
enabling 24-5
host database 24-5
overview 24-5
shunned hosts, releasing 24-7
shunned hosts, viewing 24-6
shunning attackers 24-5
system performance 24-5
targets, viewing 24-7
scanning statistics
enabling 24-7
system performance 24-7
viewing 24-8
time exceeded, ICMP message C-15
time ranges, access lists 18-19
timestamp, including in system log messages 44-20
timestamp reply, ICMP message C-15
timestamp request, ICMP message C-15
TLS1, used to access the security appliance 39-3
TLS Proxy
applications supported by ASA 27-3
Cisco Unified Presence architecture 27-59
configuring for Cisco Unified Presence 27-62
debugging for Cisco Unified Presence 27-64

licenses 27-4
tocken bucket 25-2
toolbar, floating, WebVPN 39-58
traffic flow
routed firewall 17-1
transparent firewall 17-11
traffic shaping
overview 25-4
Transform 29-12
transform set
creating 34-4
definition 29-12
transmit queue ring limit 25-2, 25-3
transparent firewall
about 17-7
ARP inspection
about 28-1
enabling 28-2
static entry 28-2
data flow 17-11
DHCP packets, allowing 18-6
guidelines 17-9
H.323 guidelines 17-8
HSRP 17-8
MAC address timeout 28-4
MAC learning, disabling 28-4
Management 0/0 IP address 8-5
management IP address 9-5
multicast traffic 17-8
packet handling 18-6
static bridge entry 28-3
unsupported features 17-10
VRRP 17-8
transparent mode
NAT 19-3
traps, SNMP 44-2
troubleshooting
H.323 26-44
H.323 RAS 26-45
phone proxy 27-26
SIP 26-74
trunk, 802.1Q 6-7
trunk ports 5-11
trustpoint 41-3
trustpoint, ASA 5505 client 36-7
trust relationship
Cisco Unified Mobility 27-56
Cisco Unified Presence 27-61
tunnel
ASA 5505 as Easy VPN client 36-5
IPsec 29-11
security appliance as a tunnel endpoint 29-1
tunnel group
ASA 5505 as Easy VPN client 36-7
configuring 32-6
creating 32-8
default 29-11, 32-1, 32-2
default, remote access, configuring 32-7
default LAN-to-LAN, configuring 32-16
definition 32-1, 32-2
general parameters 32-3
inheritance 32-1
IPSec parameters 32-4
LAN-to-LAN, configuring 32-16
name and type 32-8
remote access, configuring 34-5
remote-access, configuring 32-8
tunnel-group
general attributes 32-3
tunnel-group ISAKMP/IKE keepalive settings 32-4
tunneling, about 29-1
tunnel mode 30-2
tx-ring-limit 25-2, 25-3

U

UDP
connection limits per context 7-6

connection state information 1-18
ports and literal values C-11
unreachable, ICMP message C-15
UR (unrestricted) license 15-3
url-list
group policy attribute for Clientless SSL VPN 32-69
username attribute for Clientless SSL VPN 32-85
URLs
context configuration, changing 7-16
context configuration, setting 7-9
filtering, about 22-4
filtering, configuration 22-6
user, VPN
definition 32-1
remote access, adding 34-4
user access, restricting remote 32-80
user authentication, group policy 32-51
user EXEC mode
accessing 2-5
prompt B-2
username
adding 14-7
clientless authentication 35-9
encrypted 14-8
management tunnels 36-9
password 14-8
WebVPN 39-74
Xauth for Easy VPN client 36-4
username attributes
access hours 32-77
configuring 32-75, 32-77
group-lock 32-80
inheritance 32-77
password, setting 32-76
password-storage 32-81
privilege level, setting 32-76
simultaneous logins 32-78
vpn-filter 32-79
vpn-framed-ip-address 32-79
vpn-idle timeout 32-78
vpn-session-timeout 32-79
vpn-tunnel-protocol 32-80
username attributes for Clientless SSL VPN
auto-signon 32-87
customization 32-83
deny message 32-84
filter (access list) 32-85
homepage 32-83
html-content-filter 32-82
keep-alive ignore 32-87
port-forward 32-86
port-forward-name 32-87
sso-server 32-88
svc 32-89
url-list 32-85
username configuration, viewing 32-76
username webvpn mode 32-81
U-turn 29-19

V

VeriSign, configuring CAs example 41-5
viewing logs 44-8
viewing QoS statistics 25-13
viewing RMS 43-22
virtual cluster 31-6
IP address 31-6
master 31-6
virtual firewalls
See security contexts
virtual HTTP 21-3
virtual reassembly 1-15
VLAN mapping 32-43
VLANs 6-7
802.1Q trunk 6-7
allocating to a context 7-7
ASA 5505
configuring 5-5

MAC addresses 5-4
maximum 5-2
mapped interface name 7-8
subinterfaces 6-7
VoIP
proxy servers 26-69
troubleshooting 26-44
VPN
address pool, configuring 34-4
address pool, configuring (group-policy) 32-60
address range, subnets C-4
parameters, general, setting 31-1
setting maximum number of IPSec sessions 31-3
VPN attributes, group policy 32-41
VPN Client, IPsec attributes 29-2
vpn-filter username attribute 32-79
vpn-framed-ip-address username attribute 32-79
VPN hardware client, group policy attributes 32-50
vpn-idle-timeout username attribute 32-78
vpn load balancing
See load balancing 31-6
vpn-session-timeout username attribute 32-79
vpn-tunnel-protocol username attribute 32-80
VRRP 17-8

W

WCCP 11-9
web browsing with WebVPN 39-77
web caching 11-9
web clients, secure authentication 21-5
web e-Mail (Outlook Web Access), Outlook Web Access 39-51
WebVPN
assigning users to group policies 39-21
authenticating with digital certificates 39-21
CA certificate validation not done 39-2
client application requirements 39-75
client requirements 39-75
for file management 39-77
for network browsing 39-77
for port forwarding 39-78
for using applications 39-78
for web browsing 39-77
start-up 39-76
configuring
e-mail 39-50
configuring WebVPN and ASDM on the same interface 39-4
cookies 39-7
defining the end-user interface 39-56
definition 39-1
digital certificate authentication restrictions 39-6
e-mail 39-50
e-mail proxies 39-50
enable cookies for 39-78
end user set-up 39-56
establishing a session 39-3
floating toolbar 39-58
group policy attributes, configuring 39-22
hosts file 39-45
hosts files, reconfiguring 39-46
HTTP/HTTPS proxy, setting 39-7
Java object signing 39-53
PDA support 39-50
printing and 39-76
remote system configuration and end-user requirements 39-76
security preautions 39-2, 39-8
security tips 39-75
setting HTTP/HTTPS proxy 39-4
SSL/TLS encryption protocols 39-6
supported applications 39-75
supported browsers 39-76
supported types of Internet connections 39-76
troubleshooting 39-44
unsupported features 39-3
URL 39-76

use of HTTPS 39-3
username and password required 39-76
usernames and passwords 39-74
use suggestions 39-56, 39-75
WebVPN, Application Access Panel 39-57
webvpn attributes
group policy 32-65
welcome message, group policy 32-46
WINS server, configuring 32-40

X

Xauth, Easy VPN client 36-4

Z

Zone Labs firewalls 32-62
Zone Labs Integrity Server 14-17