
Check Point Software Technologies LTD.

FireWall-1 Version 3.0 Network to Network Encryption Quick Reference Guide With Address Translation Addendum

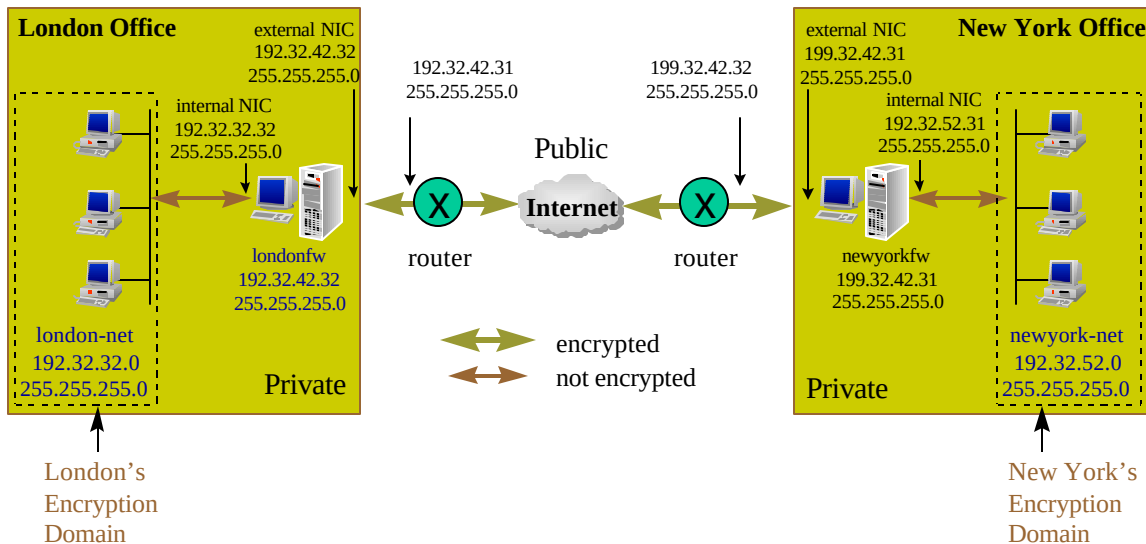
Authored By: David Goodman
Date: November 16, 1997
Purpose: To describe and Document network to network encryption (Virtual Private Network VPN) and address translation within Checkpoint Version 3.0 Firewall-1

CheckPoint Software Technologies LTD.

FireWall-1 3.0 Network to Network Encryption with Address Translation Quick Reference

Network to Network encryption Example 1

Figure 1 - Network Diagram



Given Figure 1 shown above, the goal of this exercise is to set up firewall to firewall encryption that allows networks internal to the firewall to send and receive protected data over public space.

Note: londonfw and newyorkfw must have valid Internet IP addresses to communicate over public space. Please note the network addresses of the newyork and london firewalls.

We will perform the following steps in setting up our Virtual Private Network (VPN)

- Step 1. Define the london-net and newyork-net network objects on each gateway (pg. 5)
Define the londonfw and newyorkfw firewall objects on each gateway
- Step 2. Specify encryption domains for locally defined firewalls. (pg. 9) This involves:
 - a. Defining encryption domain
 - b. Specifying FWZ as the encryption method in the firewall object properties
 - c. Editing the FWZ encryption properties: i.e. generating a CA and DH key.
 - d. Key update
- Step 3. Specify encryption domains for remotely defined firewalls. (pg. 12) This involves:
 - a. Defining encryption domain
 - b. Specifying FWZ as the encryption method in the firewall object properties
 - c. Editing the FWZ encryption properties i.e. set the property to "remote" and retrieving the public keys from the peer gateway
- Step 4. Add VPN rule to rule base (pg. 14)
- Step 5. Define FWZ encryption properties for VPN rule (pg. 15)
- Step 6. Install Security Policy (pg. 15)
- Step 7. Test the implementation (pg. 16)

The following guide assumes:

1. The two firewall machines are physically connected properly and have basic network connectivity which allow them to exchange ICMP packets.
2. The two firewall machines have properly installed VPN versions of Firewall-1 V3.x with the correct license key.

The correct license key can be verified using the “fw printlic” command as follows:

For UNIX: /etc/fw/bin/fw printlic
For NT: \winnt\fw\bin\fw printlic

The results of this command will look something like this:

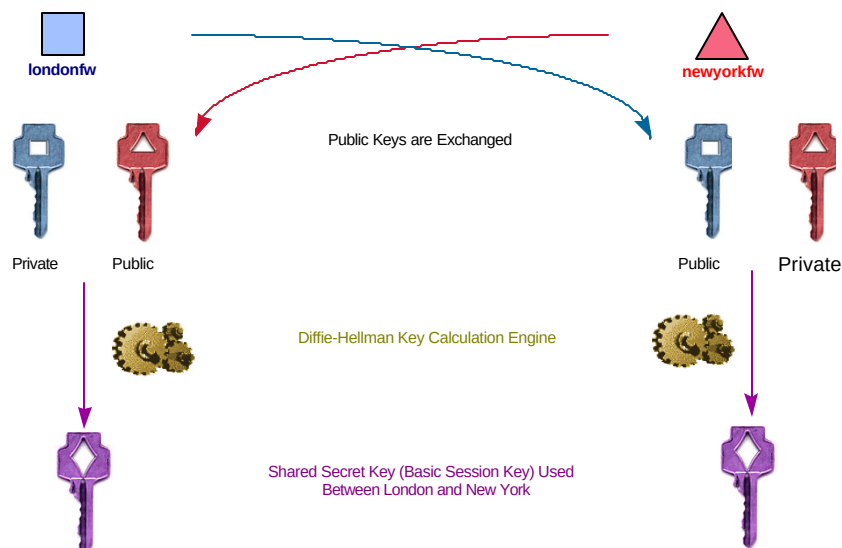
```
london# /etc/fw/bin/fw printlic
This is Firewall-1 Version 3.0a [VPN + DES]

Type           Expiration  Ver   Features
Eval           4oct97      3.x   controlx pfmx routers des skip connect
motif embedded
```

The first line may also read “This is Firewall -1 Version 3.0a [VPN]”. VPN indicates that you have the FWZ version of the software, this is a prerequisite for “firewall to firewall” or “client to firewall” encryption. The Features column lists the current licensed features of the software. In the above example, the result of the command indicates that I have the VPN version of the software, my software also contains algorithms to do DES encryption, and I am currently licensed to use des encryption & the skip encryption. **In order to do encryption, you must have the [VPN] or [VPN + DES] version of the software. Simply enabling encryption features is not enough to perform encryption.**

Encryption and key exchange

Encrypting a message modifies its text so that the encrypted text can only be read with the aid of some additional information known only to the sender and the intended recipient. This information is referred to as a secret key. The secret key can be provided in several ways but for the purposes of this exercise we will use the Diffie-Hellman Asymmetric (public key) distribution system. With this system, each correspondent has a pair of mathematically related keys: a public key known to everyone, and a private key known only to its owner. Under the Diffie-Hellman scheme, a secret key is agreed on and exchanged as follows:



London gets New York's public key and performs a calculation involving his own private key and New York's public key.

New York gets London's public key and performs a calculation involving his own private key and London's public key.

The results of both calculations are the same, and this result serves as the secret key.

Encryption Schemes

An encryption scheme consists of the following elements:

1. an encryption algorithm for encrypting messages
2. an authentication algorithm for ensuring integrity, that is, that messages have not been tampered with
3. a key management protocol for generating and exchanging keys

Firewall-1 supports three encryption schemes:

FWZ - Check Point's propriety key management, it provides full automatic key management including automated updates of Public Keys.

Manual IPSec - manual key management for IPSec.

SKIP - Sun's key management for IPSec.

For the purpose of this guide, we will use the FWZ session key encryption method and the FWZ data encryption method with no selected data integrity.

Configuring an FWZ VPN with Firewall-1

Establishing a FWZ VPN between two *FW-1* gateways requires 7 steps:

- Step 1. Define the london-net and newyork-net network objects on each gateway (pg. 5)
Define the londonfw and newyorkfw firewall objects on each gateway
- Step 2. Specify encryption domains for locally defined firewalls. (pg. 9) This involves:
 - a. Defining encryption domain
 - b. Specifying FWZ as the encryption method in the firewall object properties
 - c. Editing the FWZ encryption properties: i.e. generating a CA and DH key.
 - d. Key update
- Step 3. Specify encryption domains for remotely defined firewalls. (pg. 12) This involves:
 - a. Defining encryption domain
 - b. Specifying FWZ as the encryption method in the firewall object properties
 - c. Editing the FWZ encryption properties i.e. set the property to "remote" and retrieving the public keys from the peer gateway
- Step 4. Add VPN rule to rule base (pg. 14)
- Step 5. Define FWZ encryption properties for VPN rule (pg. 15)
- Step 6. Install Security Policy (pg. 15)
- Step 7. Test the implementation (pg. 16)

Step 1: Define the firewall and network objects on each gateway

This step will configure firewall and network objects with the following parameters:

On the london gateway

Name: londonfw
IP Address: 192.32.42.32
Net Mask: 255.255.255.0
Default
Gateway: 192.42.32.31
Location: internal
Type: gateway
Firewall-1
installed: selected (checked)

Name: newyorkfw
IP Address: 199.32.42.31
Net Mask: 255.255.255.0
Location: external
Type: gateway
Firewall-1
installed: selected (checked)

Name: london-net
IP Address: 192.32.42.0
Net Mask: 255.255.255.0
Location: internal

Name: newyork-net
IP Address: 192.32.52.0
Net Mask: 255.255.255.0
Location: external

On the newyork gateway

Name: londonfw
IP Address: 192.32.42.32
Net Mask: 255.255.255.0
Default
Gateway: 199.32.42.32
Location: external
Type: gateway
Firewall-1
installed: selected (checked)

Name: newyorkfw
IP Address: 199.32.42.31
Net Mask: 255.255.255.0
Location: internal
Type: gateway
Firewall-1
installed: selected (checked)

Name: london-net
IP Address: 192.32.42.0
Net Mask: 255.255.255.0
Location: external

Name: newyork-net
IP Address: 192.32.42.32
Net Mask: 255.255.255.0
Location: internal

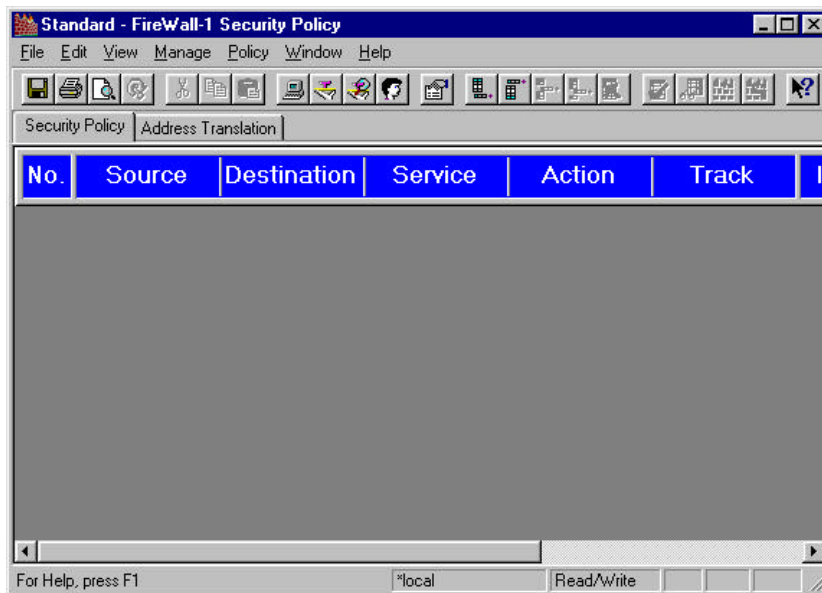
Table 1

Launch the firewall-1 security policy on the **london** gateway.

For a **Windows NT** machine the menu sequence is **Start → Programs → Firewall-1 → Security Policy**

For a **UNIX** machine the command is **/etc/fw/bin/fwpolicy**

Historical Note: “fwpolicy” has replaced the open look policy command “fwui”. If fwpolicy is not present on your machine, install the GUI client software.

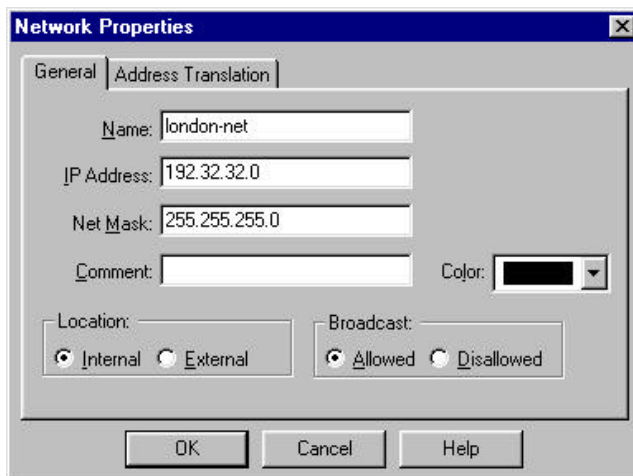


Now select **manage** → **network objects** from the main menu to display the network objects window.



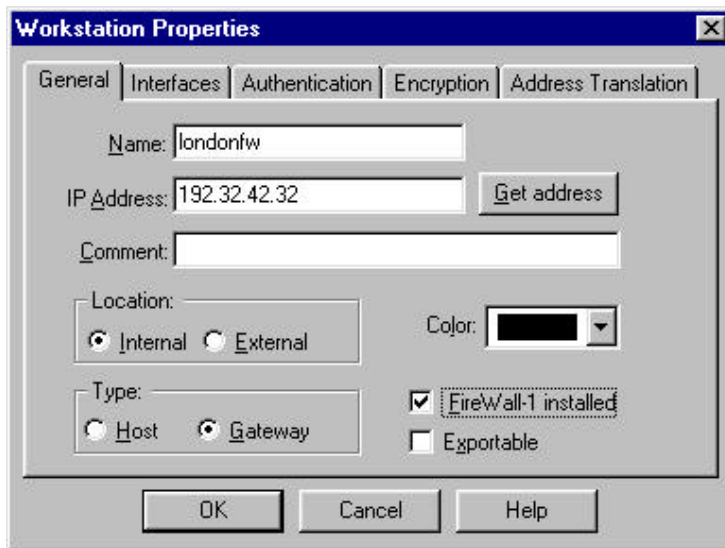
For the london gateway, select **new** → **network** and create a “**london-net**” and a “**newyork-net**” network object. This will be the set of network addresses behind the firewall between which encryption will take place. Define the “**london-net**” object with an **IP Address of 192.32.32.0** and a **network mask of 255.255.255.0**, and define the “**newyork-net**” object with an **IP Address of 192.32.52.0** and a **network mask of 255.255.255.0**.

Network objects are created before the firewall objects because they are used in the encryption domain definition of the firewall object.



Note: Be sure to specify the “**london-net**” object with an **internal** location and the “**newyork-net**” object with an **external** location on the **londonfw** gateway.

Be sure to specify the “**london-net**” object with an **external** location and the “**newyork-net**” object with an **internal** location on the **newyorkfw** gateway.



For the **london** gateway, select **new** → **workstation** and create a “**londonfw**” workstation object. Define the “**londonfw**” object with an **IP Address of 192.32.42.32**. Also create a “**newyorkfw**” workstation object. Define the “**newyorkfw**” object with an **IP Address of 199.32.42.31**.

These are the firewall definitions for this machine. If the host information of the machine is properly configured (see note below) “Get address” will fetch the proper IP address for the firewall object.

“Location:” indicates the location of the firewall relative to the gateway.

“Type:” Host specifies a workstation with a single network interface, Gateway specifies a workstation with multiple network interfaces.

The “firewall-1 installed” box is used to indicate whether firewall -1 is installed on this machine.

The “exportable” button is used in conjunction with secure remote to do client to firewall encryption. For this example we do not need to enable this feature.

Note: Be sure to specify the londonfw object with an internal location and the newyorkfw object with an external location on the londonfw gateway

Note: The firewall object must accurately define the external interface associated with the firewall module. The firewall object must have the same name as the official hostname of this interface, and the “hosts” file of the machine must contain an entry resolving the name to the proper IP address.

Check the official hostname on a UNIX or NT machine by invoking the hostname command

```
london# hostname
londonfw
```

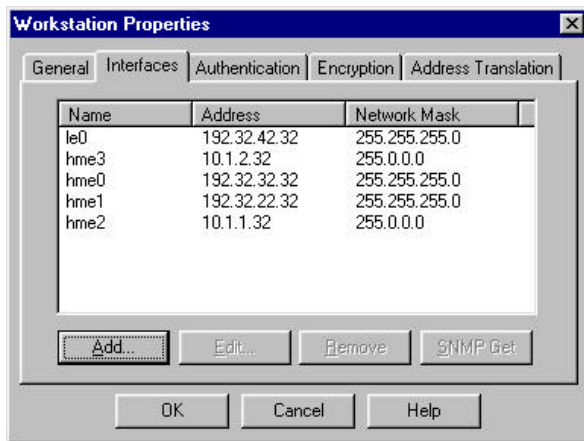
For an NT machine the “hosts” file can be found in the \winnt\system32\drivers\etc directory

For a UNIX machine the “hosts” is found in the /etc directory

check the hosts file in UNIX by invoking the more command

```
london# more /etc/hosts
#
#       internet hosts table
127.0.0.1      localhost
192.32.42.32   londonfw
192.32.32.32   bigben      #london's internal interface card
192.32.52.31   ladyliberty #newyork's internal interface card
199.32.42.31   newyorkfw
```

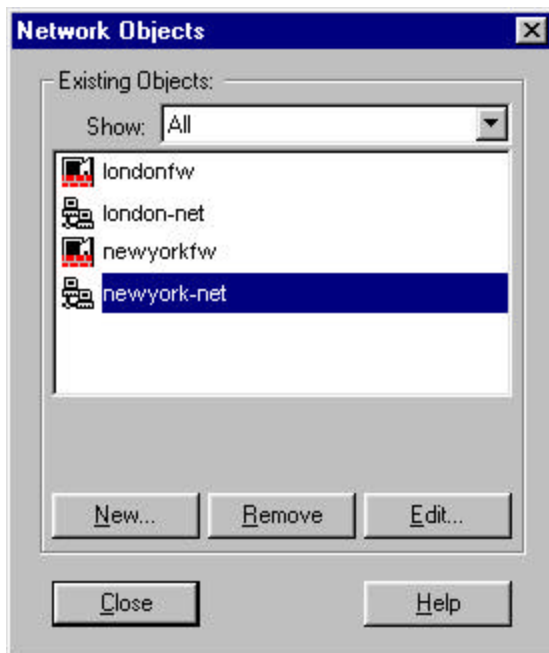
Modify the hosts file with a text editor if it does not contain the entries as shown above.



Click the **interfaces** tab in the workstation properties window and perform an **SNMP Get**.

Detailed information about the machine interfaces can be obtained by selecting the interfaces tab in the workstation properties window and selecting “SNMP Get”.

Note: You must complete this step for the security policy to be effective on all interfaces of this machine.



When the object definition step is completed, the london gateway should have a network objects window that looks like this ←

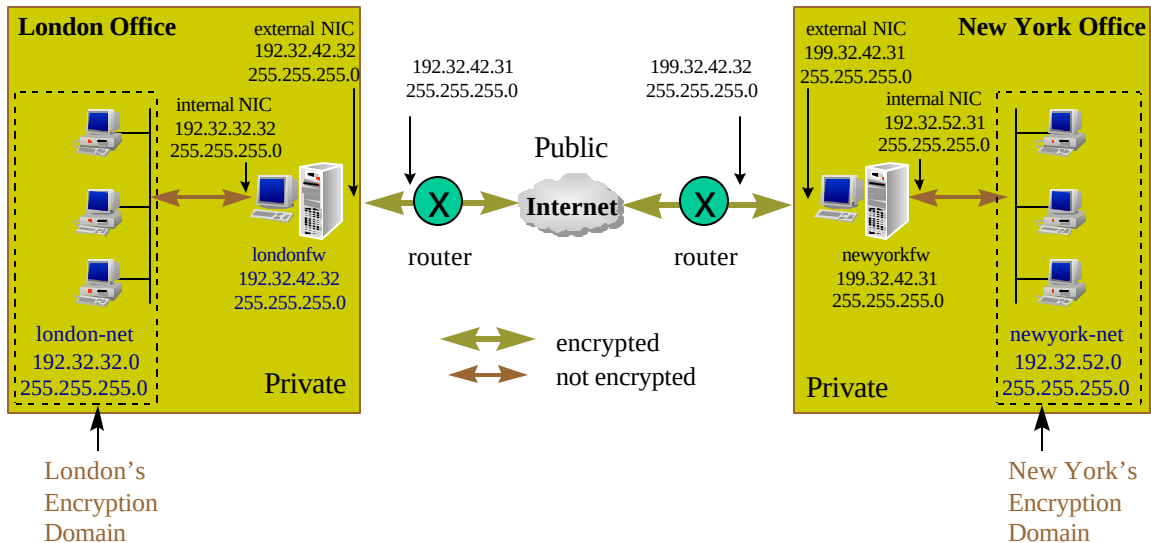
Perform a similar object setup on the newyork gateway

(Refer to the object parameters in Table 1 on page 5.)

Note: Be sure to define the network object **london-net** with an **external location** and the network object **newyork-net** with an **internal location** on the **newyorkfw** gateway.

Note: Be sure to define the firewall object **londonfw** with an **external location** and the firewall object **newyorkfw** with an **internal location** on the **newyorkfw** gateway.

Step 2: Specify encryption domains for a locally defined firewalls



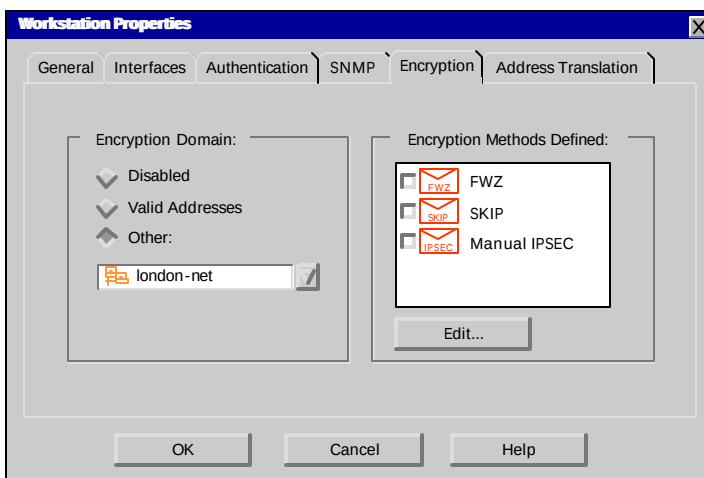
The figure above shows the intended configuration for steps 2 and 3

The objective of step 2 is to generate public keys on both gateways and make them available so the peer gateway can retrieve them in the following step 3. Four distinct activities will be performed on each gateway for the locally defined firewall object. These are:

- Define encryption domain
- Specify FWZ as the encryption method in the firewall object properties
- Edit the FWZ encryption properties: i.e. generate a CA and DH key.
- Key update

Step 2a: Define encryption domain

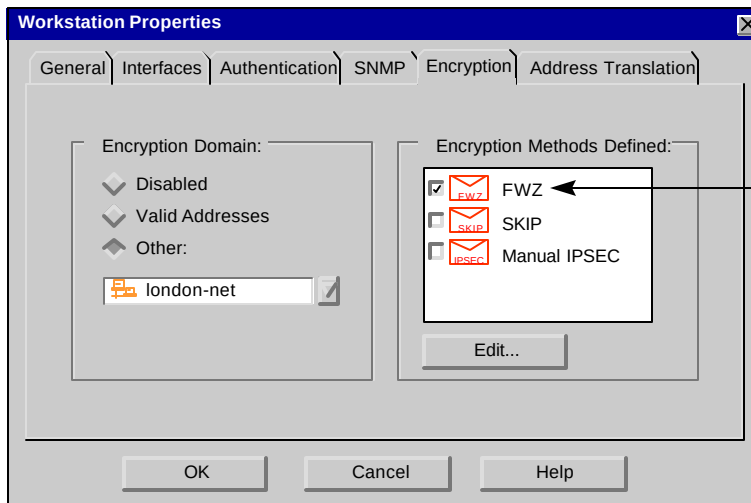
Set the encryption domain of **londonfw** firewall object on the **london** gateway as follows:



From the policy window select **manage** → **Network objects "londonfw"** → **edit** → **"Encryption"** tab. The workstation properties window will be displayed.

Select **"other"** and **london-net** for the **encryption domain** of the **londonfw** object.

Step 2b: Specify FWZ as the encryption method in the firewall object properties

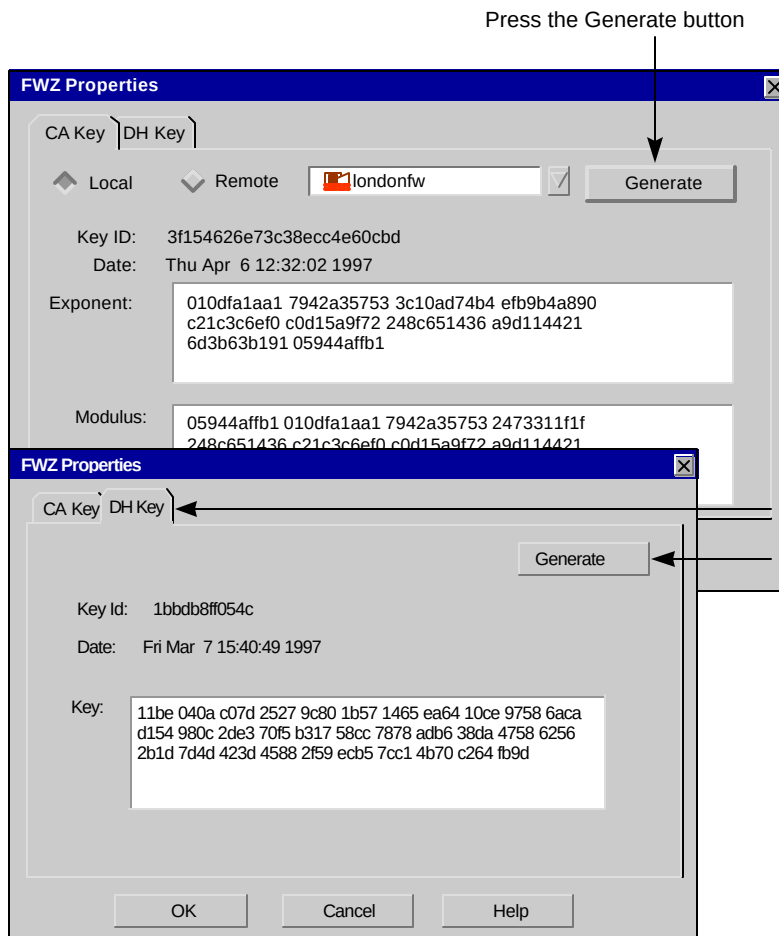


Specify the **FWZ encryption method** by selecting the radio button.

Step 2c: Edit the FWZ encryption properties: i.e. generate a CA and DH key.

The public key generation will be performed for the **londonfw** object on the **london** gateway and for the **newyorkfw** object on the **newyork** gateway.

On the **london** gateway for the **londonfw** object, select the “**FWZ**” encryption method and select **edit**. The FWZ properties window will be displayed.



Ensure the “local” button is depressed.

Click the Generate button and the FWZ CA key will be **generated** for the londonfw firewall

Select the DH Key tab at the top of the **FWZ Properties** window.

Click the Generate button displayed at the far right. This will **generate** the **FWZ Diffie-Hellman key** for your firewall.

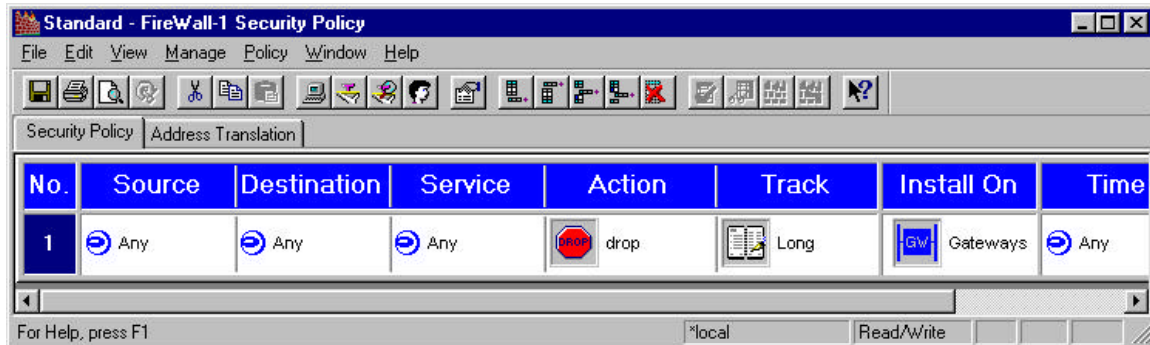
Click the OK button. The **FWZ Properties** window will disappear.

Click OK again and the workstation properties window will disappear. Click close in the

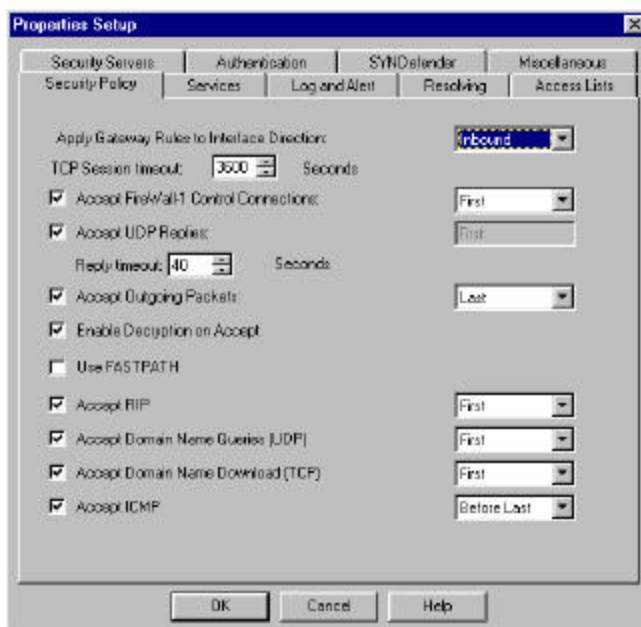
network object window

Step 2d: Key Update

At this point, a rule base must be installed in order to update the FWZ CA and DH keys on the firewall so they can be retrieved in the next step. A single rule in the rule base can achieve this goal.



From the security policy window, simply select **edit** → **add rule** → **bottom** and the implicit drop rule will be added.



Note: The firewall properties must accept firewall-1 control connections or installing this rule will deny you access to the firewall.

To check if firewall-1 control connections are accepted on your machine, **select policy** → **properties** from the **security policy** window.

The properties setup window will be displayed. In this case you can see that “**Accept Firewall-1 Control Connections:**” is enabled as part of the first rule.

Install this rule base on your local Firewall.

Select **Policy** → **install**. The install-on window will be displayed. Select your local Firewall, (in this case londonfw) and select **OK**.

Perform a similar public key generation for the local (newyorkfw) firewall object on the new york gateway

Step 3: Specify encryption domains for remotely defined firewalls.

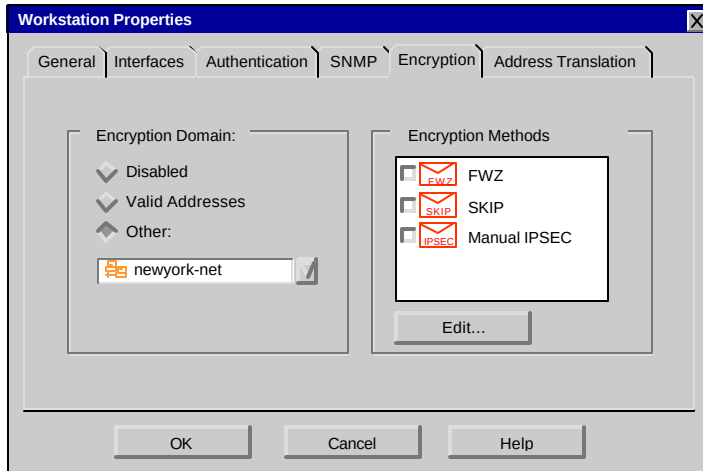
The objective of step 3 is to retrieve the public keys generated in the previous step. Three distinct activities will be performed on each gateway for the remotely defined firewall object. These are:

- a. Define encryption domain
- b. Specify FWZ as the encryption method in the firewall object properties

- c. Edit the FWZ encryption properties i.e. set the property to “remote” and retrieving the public keys from the peer gateway

Step 3a: Define encryption domain

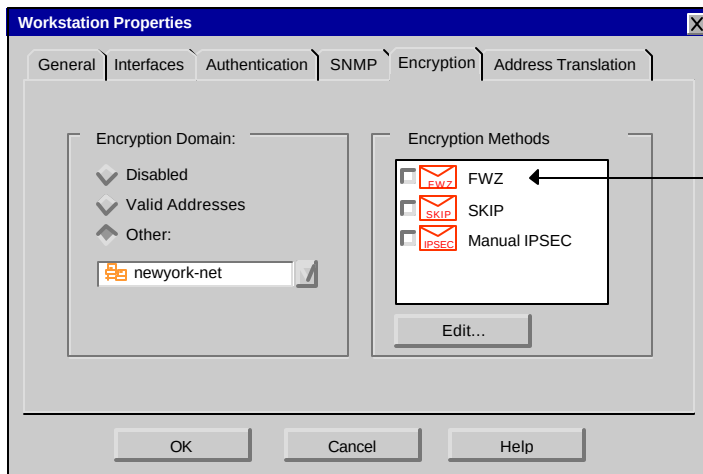
Set the **encryption domain** of the **newyorkfw** firewall object **on the london gateway** as follows:



Return to the **Network objects** window on the **london gateway** and select “**newyorkfw**” → **edit** → “**Encryption**” **tab**. The workstation properties window will be displayed.

Select “**other**” and **newyork-net** for the **encryption domain** of the **newyorkfw** object.

Step 3b: Specify FWZ as the encryption method in the firewall object properties

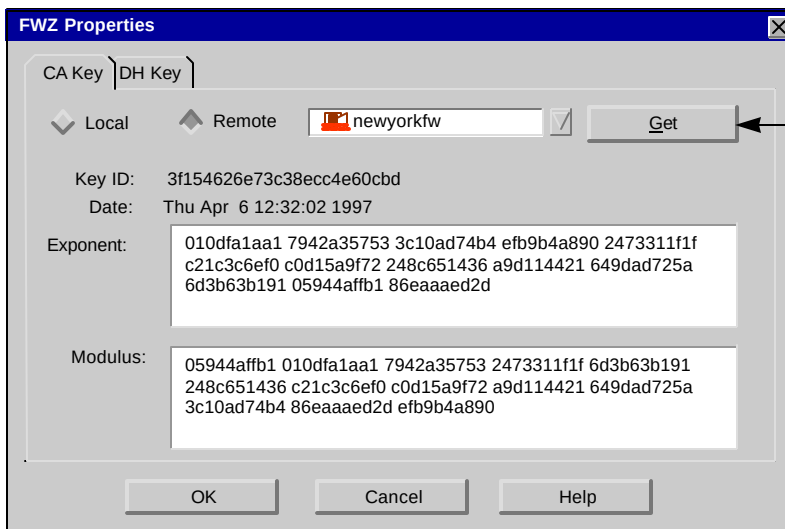


Specify the **FWZ encryption method** by selecting the radio button.

Step 3c: Exchange public keys between the gateways.

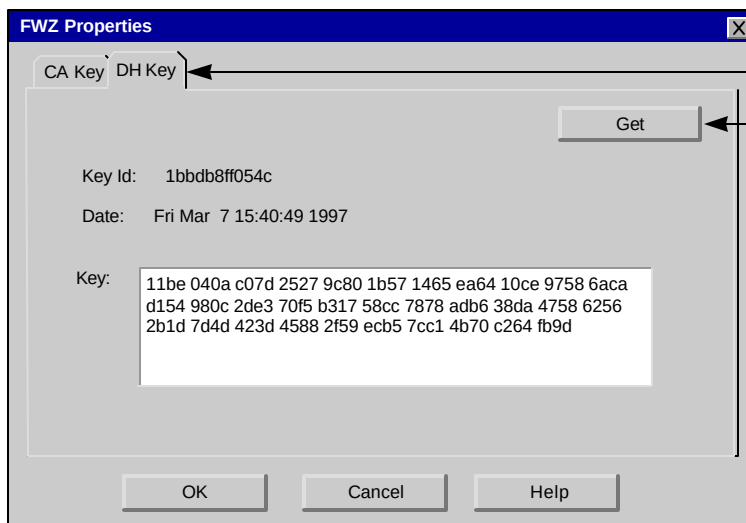
This step involves capturing the public keys generated in the previous step. The london gateway will get the newyorkfw public key and the newyork gateway will get the londonfw public key.

On the london gateway open the FWZ properties window for the newyorkfw by highlighting **FWZ** and clicking **edit** from the previous window.



Select the **Remote** button, then select **newyorkfw** from the scroll menu

Click the Get button displayed at the far right. This will retrieve the FWZ CA public key from the newyorkfw firewall.



select the **DH Key** tab at the top of the **FWZ Properties** window.

Click the Get button displayed at the far right. This will retrieve the FWZ Diffie-Hellman public key from the newyorkfw firewall.

Click the **OK** button. The **FWZ Properties** window will disappear.

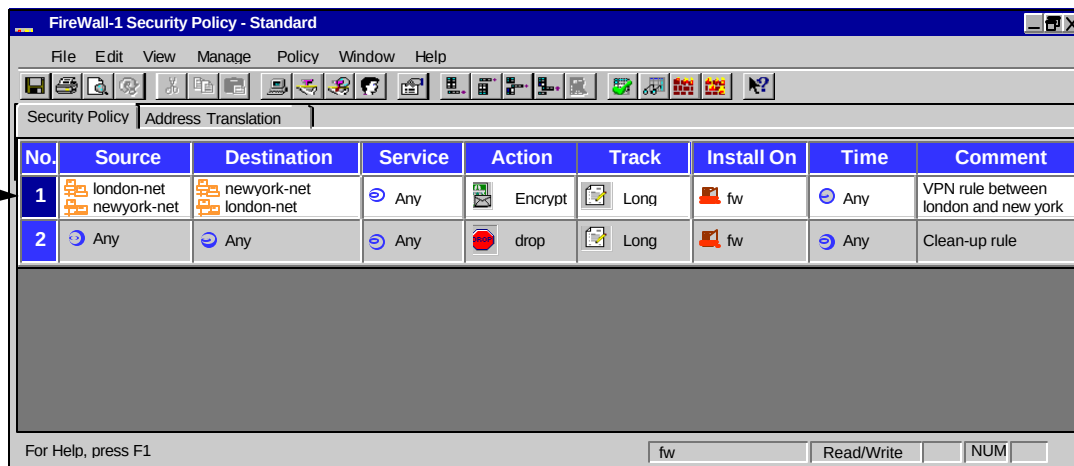
You have now acquired the newyorkfw public keys on the londonfw machine.

Note: If the Get function fails to retrieve the keys, ensure that:

1. The rule base was indeed loaded on the peer gateway: i.e. the keys are available for retrieval.
2. The two machines have IP connectivity
3. Firewall control connectivity is enabled (see the note on page 11 concerning the properties setup window)

Perform a similar public key acquisition for the remote firewall object (londonfw) on the new york gateway

Step 4: Add VPN rule to rule base



VPN rule added to top of rule base

On each gateway, add a rule to the top of the rule base

For **Destination**, specify both **london-net** and **newyork-net**.

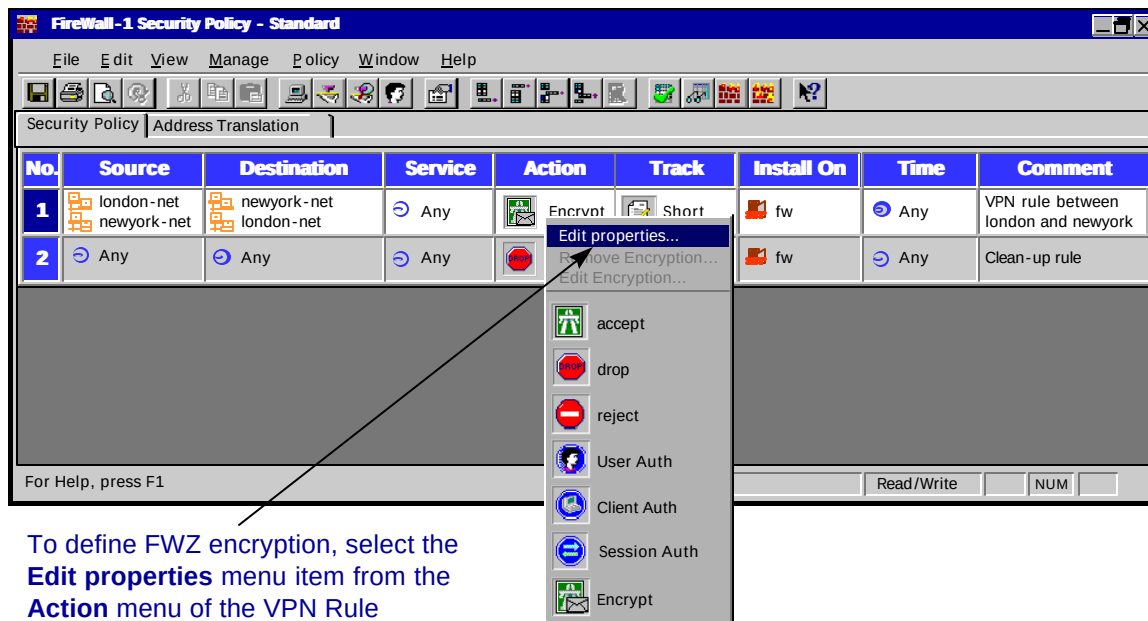
For **Source**, specify both **london-net** and **newyork-net**.

For **Service**, specify **Any**.

For **Action**, specify **Encryption**.

For **Track**, specify **Long**. This way we can see the encrypted/decrypted packets being logged in the Log Viewer.

With the cursor positioned over the **Encryption** icon in the **Action** column of the new rule, hold down the **right mouse button**. In the menu that appears, select the **Edit properties...** menu item.



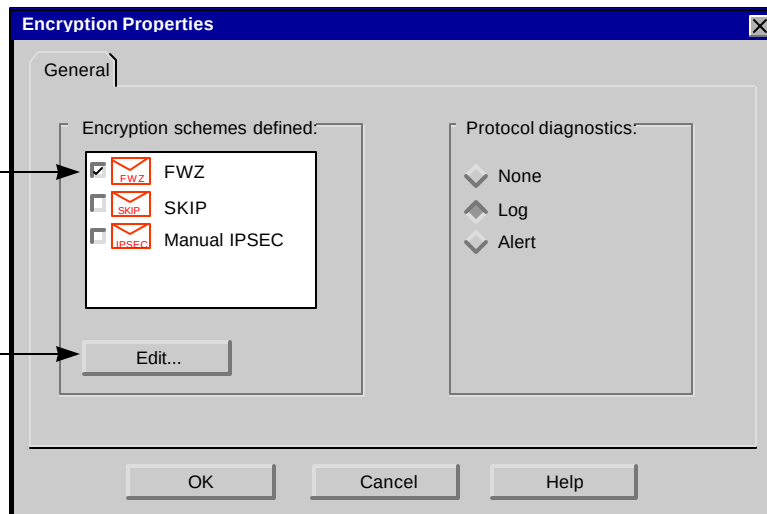
To define FWZ encryption, select the **Edit properties** menu item from the **Action** menu of the VPN Rule

The rule's **Encryption Properties** window will appear.

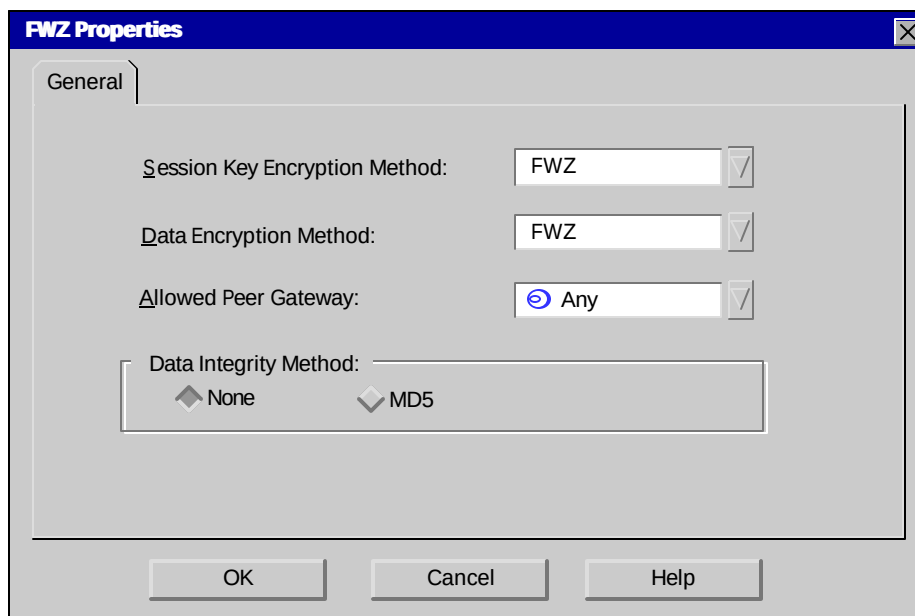
Step 5: Define FWZ encryption properties for VPN rule

In the **Encryption Properties** window, select **FWZ** as the **Encryption Method**. Select **Log** as the **Protocol Diagnostics**.

- Select **FWZ** as the encryption scheme to be used by this encryption rule
- Edit the parameters of the encryption scheme by selecting the **Edit...** button. This will invoke the **FWZ Properties** window



Select the **edit** button and the **FWZ Properties** window will be displayed.



Both firewalls in the VPN must have the same encryption settings. For the purposes of this exercise set the parameters to those shown above.

Step 6: Install Security Policy.

Verify and install the new rule base on both gateways

Step 7: Test the implementation

Network to Network encryption should now be operating properly. To test this implementation we will ping a host on one of the encryption domains and observe the encryption/decryption cycle of the ICMP packet. The ICMP packet will be encrypted as it leaves the originating gateway, and then it will be decrypted by the destination gateway. The destination gateway will encrypt the reply, and this reply will be decrypted by the originating gateway.

From a host on the newyork-net encryption domain (in this case 192.32.52.30), ping the network interface of the london-net encryption domain (in this case 192.32.32.32).

```

Microsoft(R) Windows 95
(C) Copyright Microsoft Corp 1981-1996.
C:\WINDOWS>ping 192.32.32.32
Pinging 192.32.32.32 with 32 bytes of data:
Reply from 192.32.32.32: bytes=32 time=678ms TTL=254
Reply from 192.32.32.32: bytes=32 time=3ms TTL=254
Reply from 192.32.32.32: bytes=32 time=3ms TTL=254
Reply from 192.32.32.32: bytes=32 time=3ms TTL=254
C:\WINDOWS>

```

Open the log viewer of each gateway by selecting **Window → log viewer** from the security policy window.

No	Date	Time	Inter.	Origin	Type	Action	Source	Destination	Proto.	Result
0	19Sep97	10:24:55	dae...	199.32.42.31	control	ctl				
1	19Sep97	10:25:01	dae...	199.32.42.31	log	encrypt	192.32.52.30	192.32.32.32	icmp	1
2	19Sep97	10:25:01	dae...	199.32.42.31	log	decrypt	192.32.32.32	192.32.52.30	icmp	1

Observe the encrypt/decrypt pair of ICMP packet of the newyork gateway.

The ICMP packet is encrypted leaving the gateway.

The encrypted response is decrypted when it is received back from the london gateway.

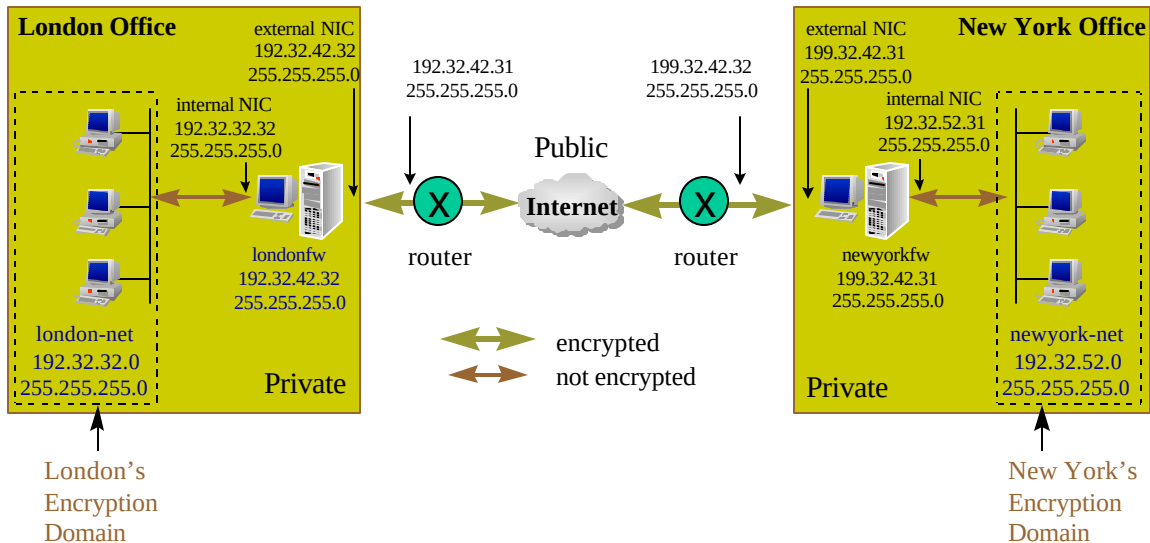
No.	Date	Time	Inter.	Origin	Type	Action	Source	Destination	Proto.	Rule	SrcKeyID	Dis.
0	19Sep97	10:24:21	des...	192.32.42.32	control	cf						
1	19Sep97	10:24:25	des...	192.32.42.32	log	decrypt	192.32.52.30	192.32.32.32	icmp	1	a2c0a6c20a80	1-4
2	19Sep97	10:24:25	des...	192.32.42.32	log	encrypt	192.32.32.32	192.32.52.30	icmp	1	14fce459526e	a2

Observe the decrypt/encrypt pair of ICMP packet of the london gateway.

The ICMP packet is decrypted entering the gateway.

The response is encrypted as it is sent back to the newyork gateway.

ADDRESS TRANSLATION WITH ENCRYPTION ADDENDUM



Given the network and encryption configuration presented in the previous section, the goal of this exercise is to configure address translation such that:

1. A client with an illegal IP address on the london-net can access the internet with a legal IP address.
2. A client with an illegal IP address on the newyork-net can access the internet with a IP legal address.
3. Traffic bound for the legal london-net client IP address is properly delivered to the illegal client IP address.
4. Traffic bound for the legal newyork-net client address is properly delivered to the illegal client IP address.
5. Traffic between the illegally addressed london-net client and the illegally addressed newyork-net client is properly encrypted and decrypted.

Note: This exercise presents a specific view of the address translation features of Firewall-1 Version 3.0 as they relate to this example. For a more detail explanation of address translation please see the document "Firewall-1 Version 3.0 Address Translation Quick Reference Guide" by Joe DiPietro dated July 11,1997.

We will perform the following steps in setting up our Address Translation operations:

- Step 1. Define the illegally addressed workstation objects on the london-net and newyork-net networks (pg. 20)
- Step 2. Identify legal IP addresses to represent the illegally addressed clients. Define these valid external legal workstation objects on each gateway. (pg. 20)
- Step 3. Set up an "FW_SRC_STATIC" address translation rule on each gateway to translate the invalid internal source address to a valid external source address in the outbound direction. (pg.21)
- Step 4. Add the proper ARP entry into the firewall. (pg. 22)
- Step 5. Set up an "FW_DST_STATIC" address translation rule on each gateway to translate the valid source address to an invalid destination address in the inbound direction. (pg. 23)
- Step 6. Define a simple rule statement to test the address translation segment.(pg. 24)
- Step 7. Install this Policy and test the address translation implementation (pg. 25)
- Step 8. Modify the policy to add encryption (pg. 25)
 - a. Create encryption domain groups that include the new legal IP addresses

- b. Define the encryption scheme and create/exchange keys for the NYvalid_external and the Lonvalid_external objects
- c. Modify the encryption domain of the firewalls
- d. Add encryption rules to the firewalls

Step 9. Test the implementation (pg. 30)

In addition to the following firewall and network objects which have been defined in the previous section:

On the london gateway

Name: londonfw
 IP Address: 192.32.42.32
 Net Mask: 255.255.255.0
 Default
 Gateway: 192.42.32.31
 Location: internal
 Type: gateway
 Firewall-1
 installed: selected (checked)

Name: newyorkfw
 IP Address: 199.32.42.31
 Net Mask: 255.255.255.0
 Location: external
 Type: gateway
 Firewall-1
 installed: selected (checked)

Name: london-net
 IP Address: 192.32.42.0
 Net Mask: 255.255.255.0
 Location: internal

Name: newyork-net
 IP Address: 192.32.52.0
 Net Mask: 255.255.255.0
 Location: external

On the newyork gateway

Name: londonfw
 IP Address: 192.32.42.32
 Net Mask: 255.255.255.0
 Default
 Gateway: 199.32.42.32
 Location: external
 Type: gateway
 Firewall-1
 installed: selected (checked)

Name: newyorkfw
 IP Address: 199.32.42.31
 Net Mask: 255.255.255.0
 Location: internal
 Type: gateway
 Firewall-1
 installed: selected (checked)

Name: london-net
 IP Address: 192.32.42.0
 Net Mask: 255.255.255.0
 Location: external

Name: newyork-net
 IP Address: 192.32.42.32
 Net Mask: 255.255.255.0
 Location: internal

We will define the following objects:

On the london gateway

Name: Lonvalid_external
 IP Address: 192.32.42.100
 Location: internal
 Type: host
 Firewall-1
 installed: not selected

Name: london_client
 IP Address: 192.32.32.35
 Net Mask: 255.255.255.0
 Location: internal
 Type: host
 Firewall-1
 installed: not selected

On the newyork gateway

Name: NYvalid_external
 IP Address: 199.32.42.200
 Location: internal
 Type: host
 Firewall-1
 installed: not selected

Name: NY_client
 IP Address: 192.32.52.30
 Net Mask: 255.255.255.0
 Location: internal
 Type: host
 Firewall-1
 installed: not selected

Step 1: Define the illegally addressed workstation objects on the london-net and newyork-net networks.

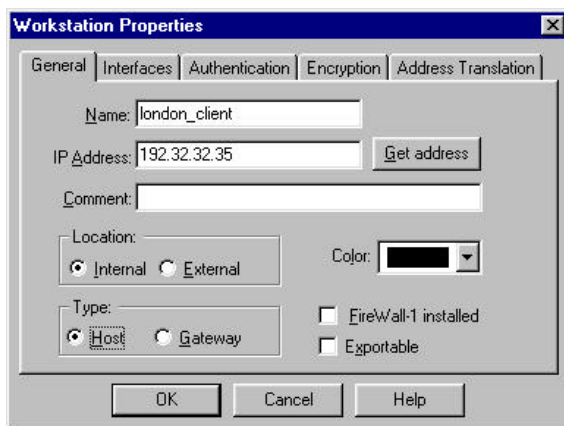
This step defines the illegally addresses internal clients with the following parameters on each machine.

On the london gateway

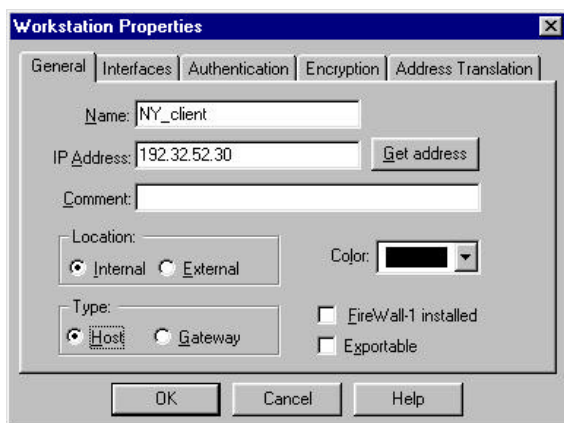
Name: london_client
IP Address: 192.32.32.35
Net Mask: 255.255.255.0
Location: internal
Type: host
Firewall-1
installed: not selected

On the newyork gateway

Name: NY_client
IP Address: 192.32.52.30
Net Mask: 255.255.255.0
Location: internal
Type: host
Firewall-1
installed: not selected



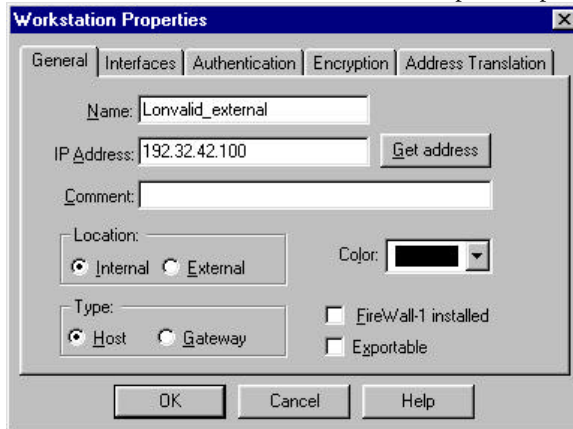
For the **london** gateway, select **new** → **workstation** and create a “**london_client**” workstation object. Define the “**london_client**” object with an **IP Address of 192.32.32.35**.



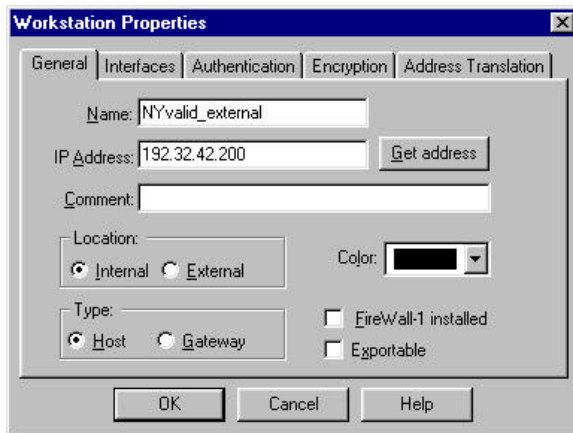
On the **newyork** gateway create a “**NY_client**” workstation object. Define the “**NY_client**” object with an **IP Address of 199.32.52.30**.

Step 2: Identify legal IP addresses to represent the illegally addressed clients. Define these valid external legal workstation objects on each gateway.

Note: For this example, we have chosen to represent illegally addressed clients behind the firewall with IP addresses outside the firewall. These external IP addresses must be valid Internet IP addresses to communicate over public space.



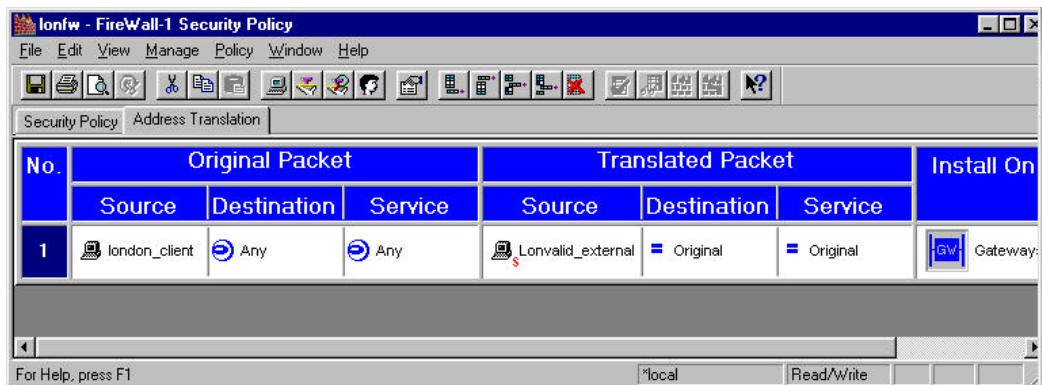
For the **london** gateway, select **new** → **workstation** and create a “**Lonvalid_external**” workstation object. Define the “**Lonvalid_external**” object with an **IP Address of 192.32.42.100**.



On the **newyork** gateway create a “**NYvalid_external**” workstation object. Define the “**NYvalid_external**” object with an **IP Address of 199.32.42.200**.

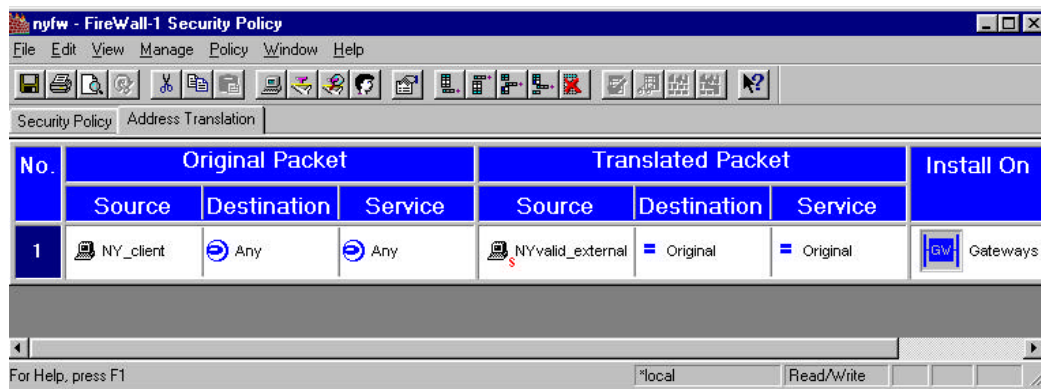
Step 3: Set up an “FW_SRC_STATIC” address translation rule on each gateway to translate the invalid internal source address to a valid external source address in the outbound direction.

On the london firewall select the “Address Translation” Tab near the “Security Policy” Tab in the main window, and enter “london_client” under the Original Packet “Source” field. Under the Translated Packet “Source” field enter “Lonvalid-external” using the “STATIC” method as shown below:



This rule entry tells the Firewall to take the original source address of 192.32.32.35 and change it to 192.32.42.100 upon exiting the external interface of the london Firewall.

On the newyork firewall select the “Address Translation” Tab near the “Security Policy” Tab in the main window, and enter “NY_client” under the Original Packet “Source” field. Under the Translated Packet “Source” field enter “NYvalid-external” using the “STATIC” method as shown below:



This rule entry tells the Firewall to take the original source address of 192.32.52.30 and change it to 199.32.42.200 upon exiting the external interface of the newyork Firewall.

Step 4: Add the proper ARP entry into the firewall.

Since the new valid external addresses cannot be resolved with an ARP request, we must make an ARP entry on the firewall to enable address resolution.

For each gateway, determine the MAC address of the external interface of the firewall.

For an NT machine this done with the command:

C:\windows\IPCONFIG /all

The results of this command are:

Windows NT IP Configuration

....
....

Ethernet adapter :

Description : le0 Ethernet Adapter
Physical Address. : 00-60-97-C9-D3-32
DHCP Enabled. : No
IP Address. : 199.32.42.31
Subnet Mask : 255.255.255.0
Default Gateway : 199.32.42.32

For a UNIX machine the command is:

ifconfig -a

Once the MAC address is determined, the ARP statement is entered in the following manner:

On the NT Platform, a file called “**local.arp**” must be created under the **\$fwdir/state directory**. This is usually located under **C:\winnt\fw\state** if the default installation parameters have been taken during the installation process. The format of this file is as follows:

IP Address

MAC Address

So in our example, the local.arp file would look like this with a “-“ between MAC address characters:

199.32.42.200 00-60-97-C9-D3-32

This means when someone is lookup for the IP address of 199.32.42.200, talk to the External Interface of the Firewall (00:60:97:C9:D3:32). This will allow the ARP Request packet for 199.32.42.200 to be resolve by the Firewall.

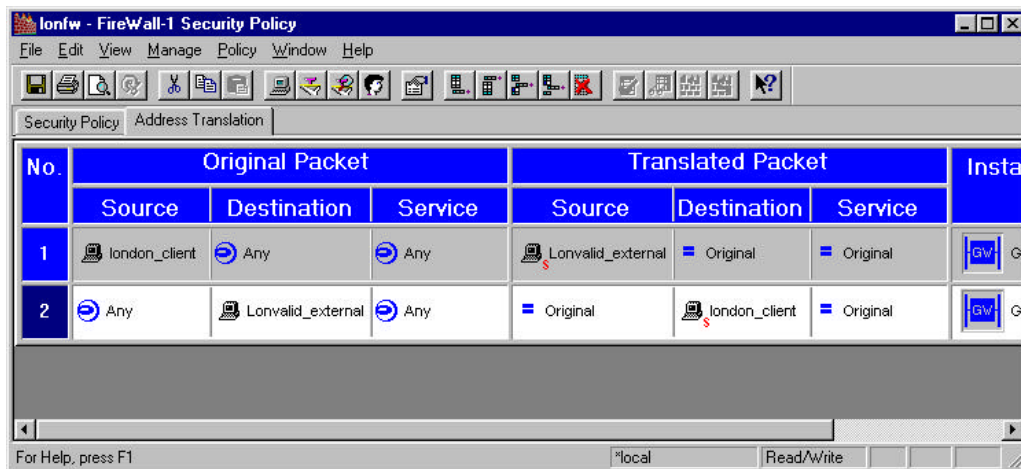
On a UNIX platform the ARP statement is entered with the command:

arp -s 199.32.42.200 00:60:97:C9:D3:32 pub

On each gateway, enter the proper ARP statement corresponding to the new legal external IP addresses.

Step 5: Set up an “FW_DST_STATIC” address translation rule on each gateway to translate the valid source address to an invalid destination address in the inbound direction.

On the london gateway, add the following entry to the Address Translation Tables as shown in Rule Number 2 with the Original Packet Destination being “Lonvalid_external” and the Translated Packet Destination “london_client”.



Add a proper routing entry in the Firewall to forward the Packet from 192.32.42.100 to 192.32.32.35

The proper syntax for each platform is shown below.

UNIX

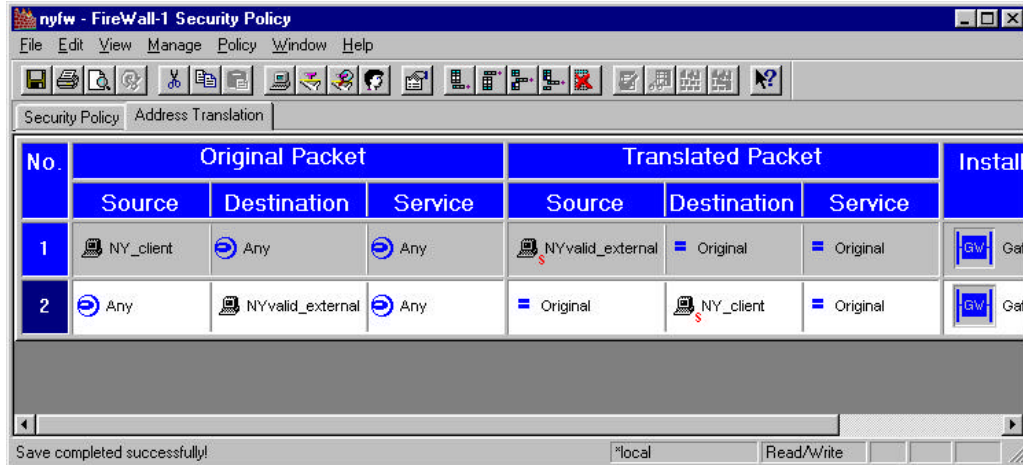
route add 192.32.42.100 192.32.32.35 1

WINDOW NT

route add -p 192.32.42.100 192.32.32.35

The -p is for permanent

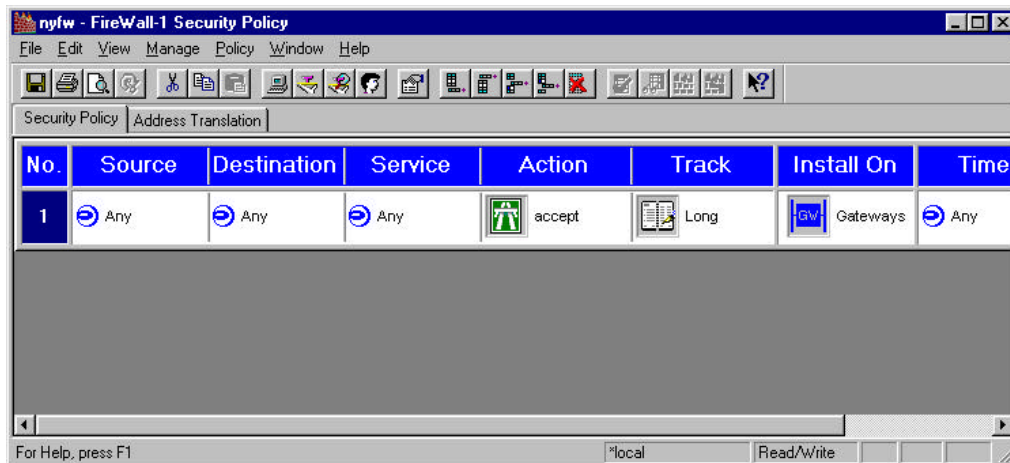
On the newyork gateway, add the following entry to the Address Translation Tables as shown in Rule Number 2 with the Original Packet Destination being “NYvalid_external” and the Translated Packet Destination “NY_client”.



Add a proper routing entry in the Firewall to forward the Packet from 199.32.42.200 to 192.32.52.30

Step 6: Define a simple rule statement to test the address translation segment.

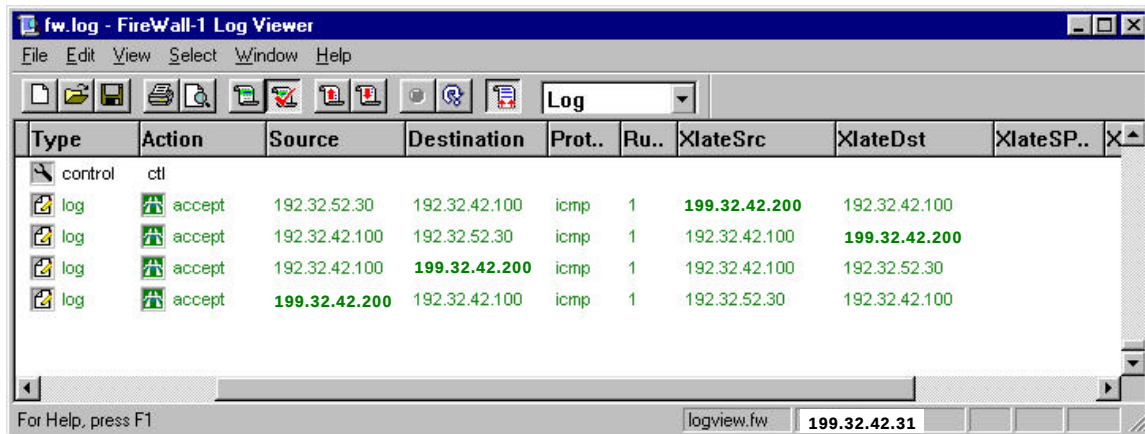
On each gateway add a rule as shown below.



Step 7: Install this Policy and test the address translation implementation.

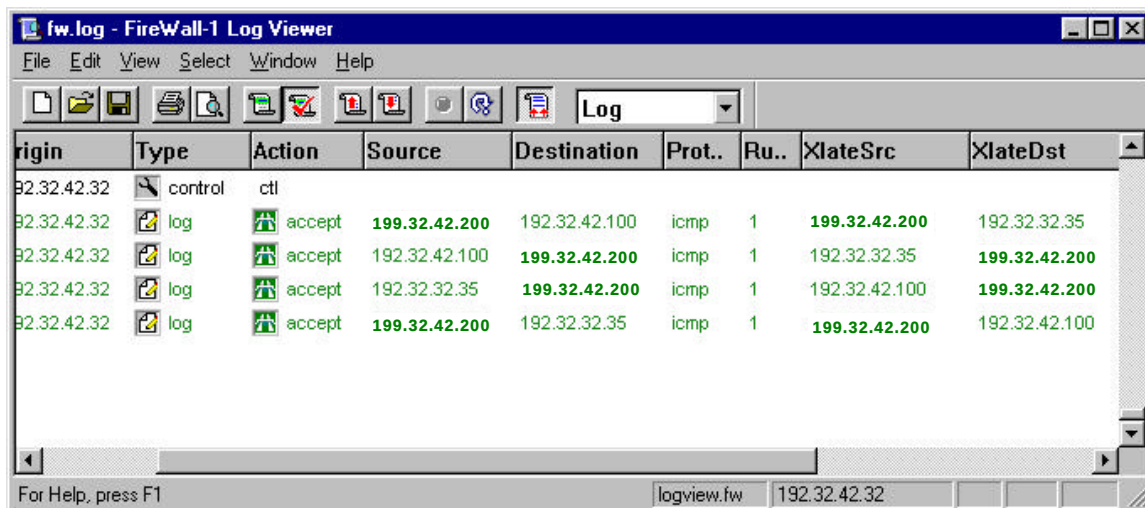
On each gateway, **Select Policy → Install** from the main window.

From the newyork gateway, **ping 192.32.42.100**. Observe the proper translation results in the log viewer.



Type	Action	Source	Destination	Prot..	Ru..	XlateSrc	XlateDst	XlateSP..
control	ctl							
log	accept	192.32.52.30	192.32.42.100	icmp	1	199.32.42.200	192.32.42.100	
log	accept	192.32.42.100	192.32.52.30	icmp	1	192.32.42.100	199.32.42.200	
log	accept	192.32.42.100	199.32.42.200	icmp	1	192.32.42.100	192.32.52.30	
log	accept	199.32.42.200	192.32.42.100	icmp	1	192.32.52.30	192.32.42.100	

From the london gateway, **ping 192.32.42.200**. Observe the proper translation results in the log viewer



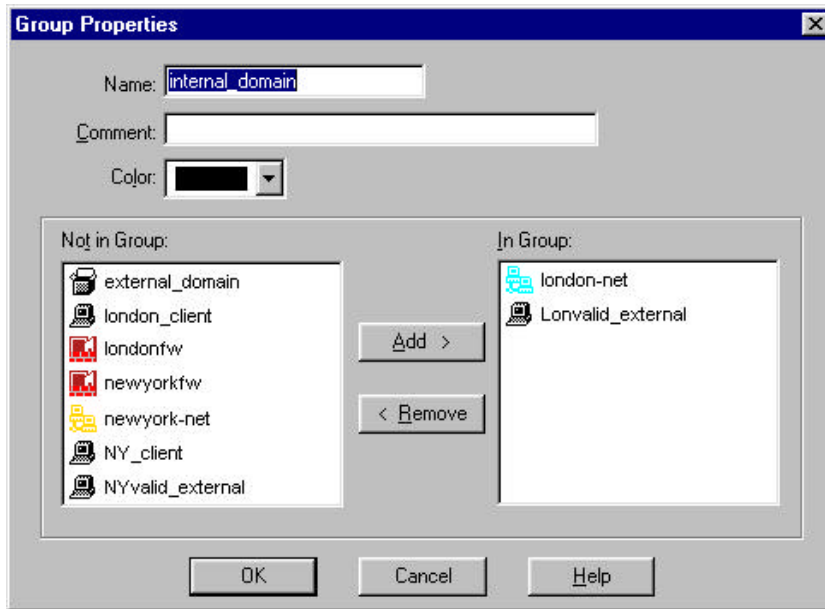
Origin	Type	Action	Source	Destination	Prot..	Ru..	XlateSrc	XlateDst
192.32.42.32	control	ctl						
192.32.42.32	log	accept	199.32.42.200	192.32.42.100	icmp	1	199.32.42.200	192.32.32.35
192.32.42.32	log	accept	192.32.42.100	199.32.42.200	icmp	1	192.32.32.35	199.32.42.200
192.32.42.32	log	accept	192.32.32.35	199.32.42.200	icmp	1	192.32.42.100	199.32.42.200
192.32.42.32	log	accept	199.32.42.200	192.32.32.35	icmp	1	199.32.42.200	192.32.42.100

Step 8 Modify the policy to add encryption

a. Create encryption domain groups that include the new legal IP addresses

At this point we must create groups on each firewall to represent the encryption domain. On the london gateway, select **manage → network objects → new → groups**

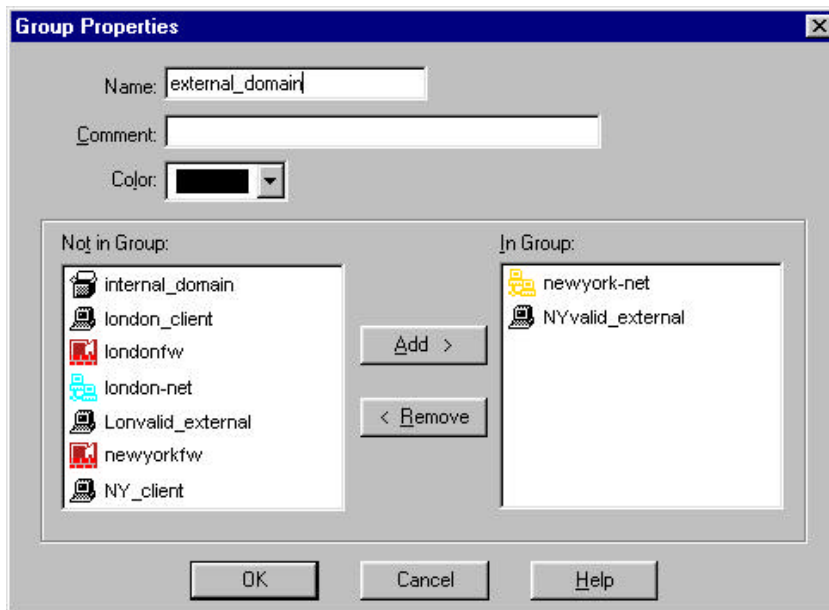
The following window will appear:



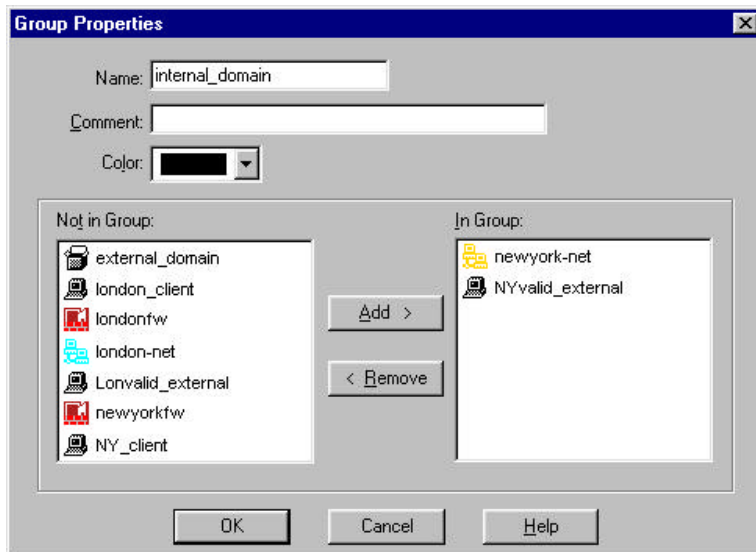
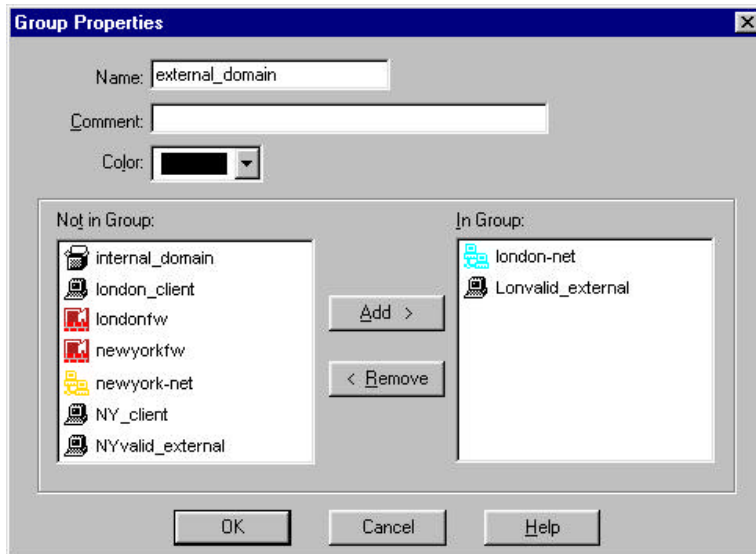
Name the group
internal_domain.

Add the Lonvalid_external
object and the london-net
object and **select OK**

Also on the london gateway create a external_domain group as shown below:



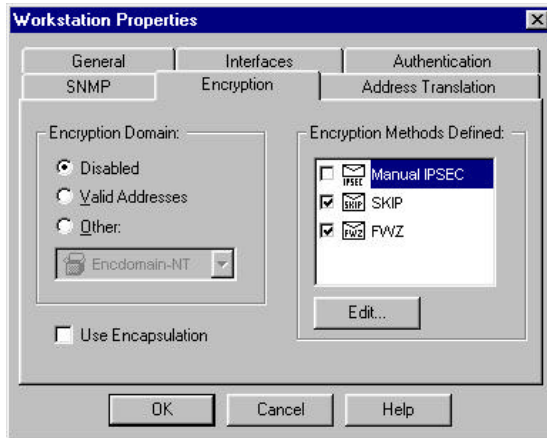
Perform the same group creation on the newyork firewall as shown:



- b. Define the encryption scheme and create/exchange keys for the NYvalid_external and the Lonvalid_external objects.

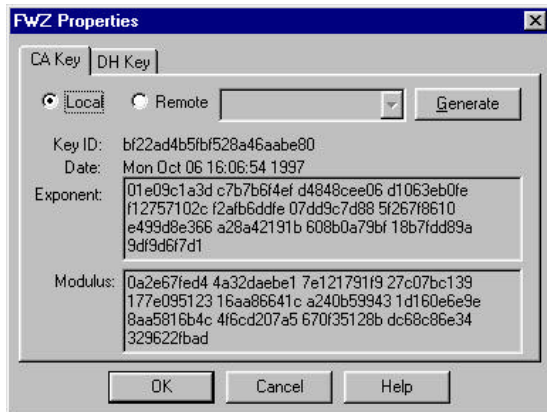
From the main menu on the london gateway, select **Manage → network objects → Lonvalid_external → edit → encryption**

the following screen will appear:

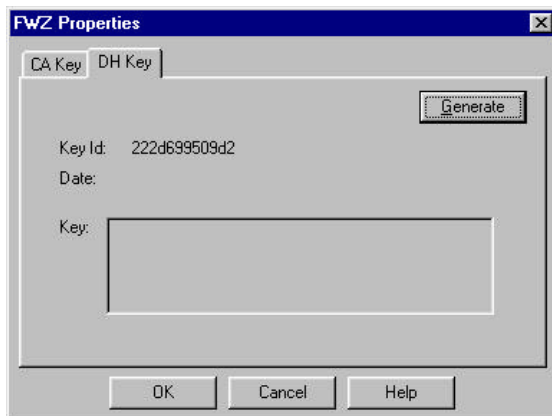


Select **FWZ → Edit**

The following screen will appear:



Notice the CA already exists. The londonfw is the Certificate Authority and it was generated in a previous section.



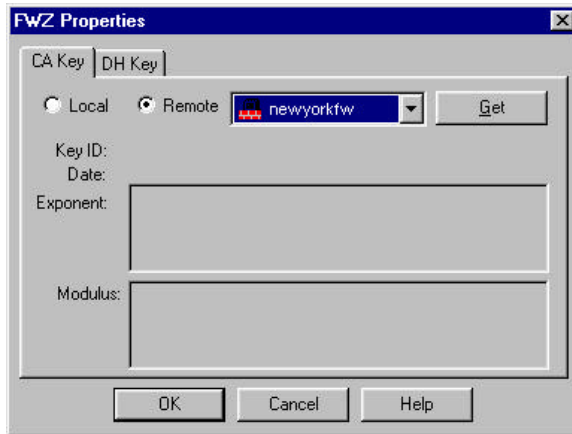
Select the **DH Key** tab and select **generate**.

Perform the same FWZ encryption scheme designation and DH Key generation for the NYvalid_external object on the newyork gateway.

At this point, the security policies must be installed on the gateways so these keys may be exchanged.

From the main menu on the london gateway, select **Manage → network objects → NYvalid_external → edit → encryption**

the following screen will appear:



Select **remote** and the newyorkfw object and **Get** the **CA and DH Keys**

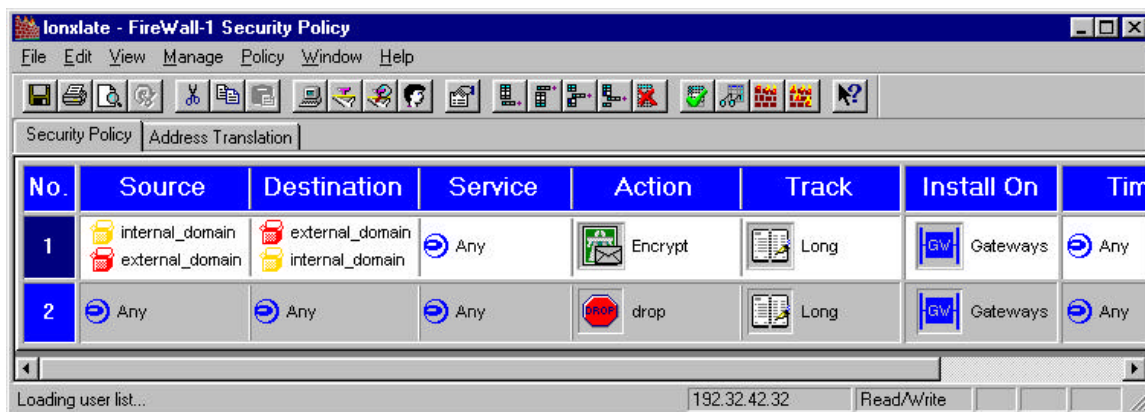
Perform the same CA and DH Key **Get** from the londonfw object for the Lonvalid_external object on the newyork gateway.

c. Modify the encryption domain of the firewalls

On each gateway, modify the encryption domains of the firewall objects to correspond to the encryption domain groups in step a. On the london gateway, the encryption domain for the londonfw is the **internal_domain** group and the encryption domain for the newyorkfw object is the **external_domain** group. On the newyork gateway, the encryption domain for the londonfw object is the **external_domain** group and the encryption domain for the newyorkfw object is the **internal_domain** group.

d. Add encryption rules to the firewalls

Produce a rule base with the following rules on each gate. Include the address translation rules configured in steps 3 and 4.



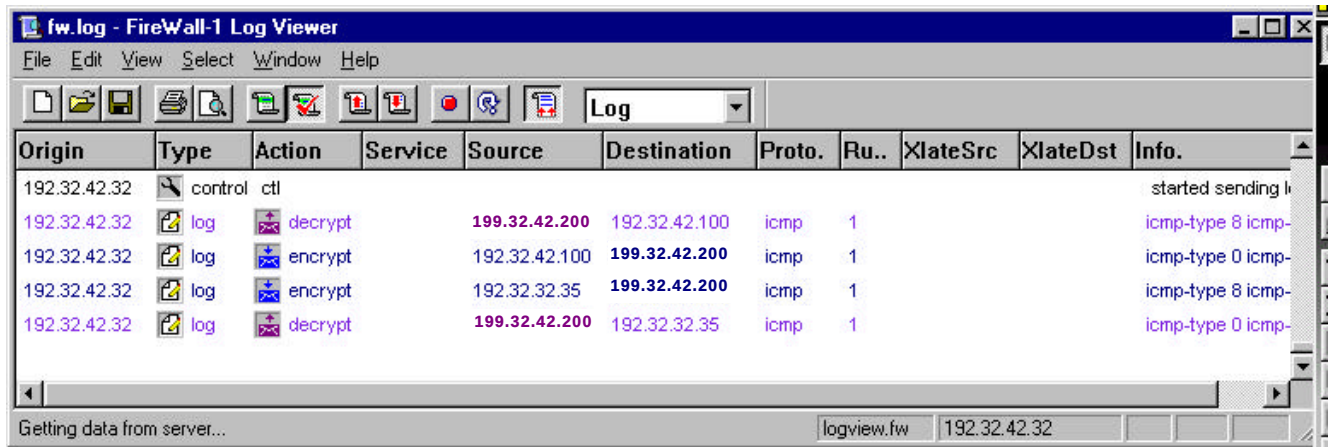
Do not forget to modify the encrypt properties (right mouse button over encrypt → Edit Properties → FWZ) in the encrypt rule.

Install the rule base on both firewalls.

Step 9 Test the implementation

Ping each valid IP address from the clients on each machine.

Observe in the log viewer the proper encryption taking place.



Note: At this time, when address translation and encryption occur at the same rule, the xlateSrc and xlateDst fields are not populated.