Chapter 9

Network Address Translation

This chapter discusses the topic of Network Address Translation: what it is, why it was created, and how you can implement it in FireWall-1. I first discuss the reasons NAT was created and how NAT is implemented in FireWall-1. Next, I show a step-by-step example of how to implement NAT in a network. I then talk about some of the inherent limitations of NAT and discuss a couple of ways to work around these limitations. Finally, I talk about troubleshooting NAT with a packet sniffer.

By the end of this chapter, you should be able to:

- Understand why NAT is necessary
- Identify what NAT actually does
- Identify why NAT does not always work
- Effectively troubleshoot NAT problems with a packet sniffer
- Implement a NAT Configuration

Introduction

Back in the old days of the Internet, the TCP/IP address space (as defined by IPv4, the version of IP used today) was thought to be more than enough. Organizations could reserve their own address space through the Internet Assigned Numbers Authority (IANA, now called the Internet Corporation for Assigned

Names and Numbers [ICANN]) and anyone who wanted a block of IP addresses generally got them.

Since the early 1990s, various people have been predicting that the IPv4 address space will simply "run out" of available addresses. This is partially due to the explosive growth of the Internet, but it is also due to how the IPv4 address space is divided. Many organizations that were allocated address space early on simply have more address space allocated to them than they are using on the Internet. There are also parts of the IPv4 address space that are not legal for hosts to be assigned to on the Internet, namely the multicast (224.0.0.0/240.0.0.0 mask) and the Class E (240.0.0.0/240.0.0.0) address spaces.

As organizations are connecting to the Internet, some are discovering that their internal network does not connect well to the Internet. The main reason for this is usually a conflict in addressing. Long before the Internet was a household word, some corporations set up their internal networks using "made up" addresses. However, you cannot simply make up addresses and use them on the Internet. You must use IP addresses assigned by IANA or an ISP. Renumbering a large, internal network would be a daunting task, not to mention that your ISP or IANA is not likely to give you enough addresses to cover all your hosts. Then again, does every host on your internal network really need to be uniquely addressable on the Internet?

IPv6 (the next version of IP) has far more address space—128 bits of address space versus the 32 provided by IPv4—which will solve this problem. However, most of today's Internet is still running IPv4 and probably will be for some time to come. A solution is needed that will help extend the IPv4 address space that is used today.

Network Address Translation (NAT) does exactly this. It is a technology that allows hosts to transparently talk to one another with addresses that are agreeable to each other. To put it another way: it allows hosts with illegal or private address space to talk with hosts on a public network and vice versa. It is a godsend for network managers who have limited address space or want to make better use of the address space they have without having to subnet, thus reducing the number of IPs that can be used. NAT can also be perceived as a security enhancement because a firewall is required for communication between the hosts. NAT, as it is commonly implemented today, is described in RFC3022.[1]

[1] You can get a copy of RFCs from www.rfc-editor.org, among other places.

NAT is implemented as part of the FireWall-1 Kernel Module that sits between the data link and network layers. As such, NAT can be provided transparently without the client or the server's knowledge. Application proxies, by their nature, can also provide this functionality as they originate all connections coming from the internal network. However, proxies usually are not transparent and do not usually give you the level of control you have over FireWall-1's NAT functionality. You can modify the source, destination, and service port of any connection going through FireWall-1.

Consider the following example (see Figure 9-1). Let's say your ISP gives you a /29 block of addresses (net mask 255.255.255.248). If you were to use this address space between your Internet router and your firewall, the address space would break down into the host numbers listed in Table 9-1.[2]

Table 9-1 Breakdown of 192.168.0.0/29 address space

Host Number	Description
.0	Network identifier (cannot be used by hosts)
.1	Internet router
.2	Firewall
.3	Available
.4	Available
.5	Available
.6	Available
.7	Broadcast address (cannot be used by hosts)

Between the broadcast address, the network address, your firewall, and your Internet router, you have a grand total of four usable IP addresses. With NAT, you can

- Give your e-mail, intranet Web server, and Web server externally reachable IP addresses

[2] If you are unfamiliar with subnetting and how it affects address space, you might want to read *LAN Technologies Explained* by Philip Miller and Michael Cummins (ISBN#1555582346), *TCP/IP Illustrated* by W. Richard Stevens (ISBN#0201633469), or any other appropriate TCP/IP book.

- Allow all your clients to access the Internet using the firewall's external IP address
- Have all of your computers protected by your firewall
- Change ISPs without having to renumber your internal network

Figure 9-1 illustrates a sample network.

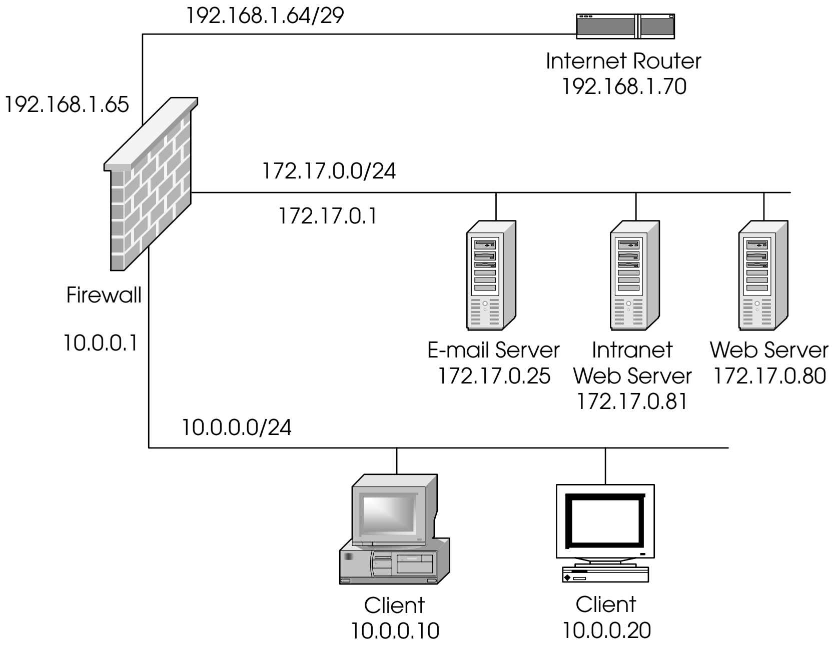


Figure 9-1 Sample network diagram

Although NAT does add an extra layer of protection and gives you flexibility, there are some downsides to NAT:

- Using NAT is like using proxies in that NAT must be updated to handle new applications. As a result, it is not compatible with every application that exists today or in the future.
- NAT requires additional work to maintain. This is discussed in more detail in the "Implementing NAT, a Step-by-Step Example" section later in this chapter.

- Only so many connections can be hidden behind a single IP address.
- NAT requires extra memory and CPU on the gateway. In most cases, this is negligible, but it starts becoming noticeable when over 20,000 connections through a single gateway are subject to NAT.

More information about the disadvantages of using NAT is documented in RFC3027.

RFC-1918 and Link Local Addresses

RFC-1918 (which was originally described in RFC-1597) sets aside specific ranges of IP addresses that cannot be used on the Internet. Instead, these addresses are to be used internally within an organization or network. If hosts with RFC-1918 addresses want to communicate with a network like the Internet, they must go through some form of NAT as no host on the Internet will know how to route RFC-1918 addresses. The addresses assigned by RFC-1918 are as follows:

- 10.0.0.0/8 (net mask 255.0.0.0)
- 172.16.0.0/12 (net mask 255.240.0.0, which covers 172.16.0.0 - 172.31.255.255)
- 192.168.0.0/16 (net mask 255.255.0.0)

Another set of address space that can be used for NAT is 169.254/16 (net mask 255.255.0.0). This address space is specified in an Internet Draft called "Dynamic Configuration of IPv4 link-local addresses," which is available at www.ietf.org/internet-drafts/draft-ietf-zeroconf-ipv4-linklocal-02.txt. Essentially, Microsoft Dynamic Host Configuration Protocol (DHCP) clients use this method to assign an address when they are unable to communicate with a DHCP server. This address space is reserved specifically for this purpose, so it will not be in use anywhere on the Internet and is thus safe to use for NAT.

If your situation requires the use of NAT, it is highly recommend that you use address space within the recommended ranges. If you are using someone else's address space within your internal network and you need to communicate with an Internet host that happens to use the same address range, you may find yourself not being able to do so as the network traffic may never leave your internal network.

How NAT Works in FireWall-1

NAT is configured via the Address Translation tab in the Security Policy Editor. The rules are processed in the order in which they are listed. Once a rule "matches" a packet, the packet is translated, and no further processing occurs. If a packet does not match a rule in the address translation rules, the packet is not translated.

Four types of NAT are available in FireWall-1, and they can be mixed and matched as necessary: Source Static, Destination Static, Destination Port Static, and Source Hide:

Source Static: Translates the source IP address in an IP packet to a specific IP address. This is a one-to-one address translation. Return traffic, as necessary, is allowed back through without additional NAT rules. However, if you need to initiate connectivity from either side of the firewall, a corresponding Destination Static NAT rule is needed.

Source Hide: Makes more than one host appear as a single host (i.e., a many-to-one translation). In the text, I will refer to this simply as *hide mode*. This is perfect for hosts that require access to the Internet, but should not be accessed *from* the Internet. In order to accomplish this, FireWall-1 changes the source TCP or UDP port of the packet so that it can keep track of which host the connection belongs to (and, consequently, know where to send reply packets). For ICMP packets, the data portion of the packet is munged (the data portion of an ICMP packet usually isn't used). For other IP protocols, hide mode does not work because there are no ports or data that can be modified. Most standard applications (e.g., Telnet, HTTP, FTP, HTTPS) work fine, but any application that requires a connection initiated from the outside or requires that a connection happen on a specific source port will not work in hide mode. An example of such is Internet Key Exchange (IKE) as used with IPSec implementations.

Destination Static: Translates the destination IP address in an IP packet to a specified IP address. This is a one-to-one address translation for connections. Return traffic, as necessary, is allowed back through without additional NAT rules. However, if you need to initiate connectivity from either side of the firewall, a corresponding Source Static NAT rule is needed.

Destination Port Static: Translates only the destination (or service) port number to a different port. This, for example, allows you to transparently

> request going from port 8080 to port 80. However, it does *not* allow you to make services running on machines other than the firewall accessible with the firewall's IP address; that is, if you want to make services accessible through the firewall, a different routable IP address is needed.[3]

NAT rules apply to all interfaces and cannot be applied on a per interface basis. Usually, rules can be crafted in such a way that per interface rules are not necessary. You can *hide* connections behind the IP 0.0.0.0, which is a special IP that tells FireWall-1 to use the interface the packet is routed out as opposed to a fixed IP address.[4]

Even though NAT can be configured in the Security Policy Editor, you need to configure the host OS, as described in the next section, in order to support NAT.

The Order of Operations

In order to understand how to implement NAT, it is best to review the "order of operations" as it relates to FireWall-1 and passing traffic in general. Consider the following case where Client A wants to communicate with Client B (see Figure 9-2).

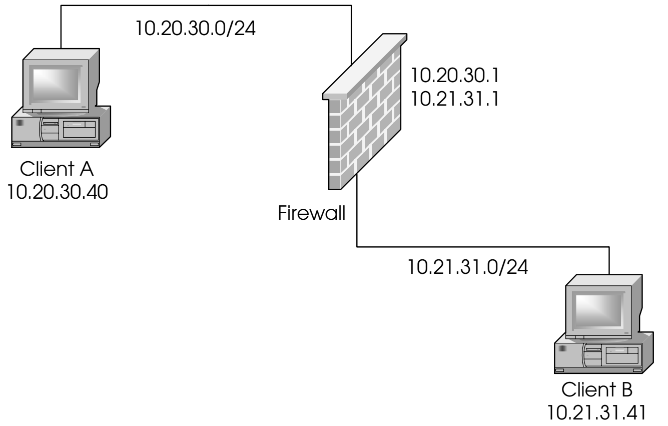


Figure 9-2 Client A communicating with Client B

[3] In order to make services accessible via the firewall's IP address, you need to use a third-party application commonly called a "plug proxy." Plug proxy simply accepts TCP or UDP packets and forwards them to a configurable host and port.

[4] Nokia platforms running Virtual Router Redundancy Protocol (VRRP) with FireWall-1 4.1 SP2 or FireWall-1 4.1 SP3 hiding behind the address 0.0.0.0 may instead use the VRRP IP address. Although some would argue this isn't necessarily bad behavior, it is not the intended behavior. FireWall-1 4.1 SP4 and later should resolve this issue.

NOTE! In this example, NAT is not configured.

Client A determines that in order to communicate with Client B, the packet must be routed through the firewall. Client A needs to know the Media Access Control (MAC) address for the firewall's IP address (10.20.30.1), so it sends out a request via the Address Resolution Protocol (ARP) requesting the address. The firewall responds with its MAC address. Client A is then able to forward the packet to the firewall for processing.

Note that all of these events happen without any aid from FireWall-It is important to be aware of this exchange because when you do address translation, you must be sure that all of the "translated" IP addresses you set up through FireWall-1 get routed back to the firewall for processing. If the translated IP address is on the same subnet as the firewall, you need to set up a proxy-ARP or static host route for that address. Otherwise, routes to those addresses will be necessary.

Once the packet is received at the firewall, FireWall-1 processes the packet according to the following steps:

1. Checks to see if the packet is part of an established connection. Because this is a new connection, there is no record of the packet in the connections table, so the connection must be checked against the security policy.
2. Checks IP Options. If the packet is denied because of this check, you will see a drop on Rule 0 in the Log Viewer, assuming that IP Options logging is enabled.
3. Performs an anti-spoofing check on the 10.20.30.1 interface. The source of the packet (10.20.30.40) is compared against the valid address setting. If the packet is denied because of this check, you will see a drop on Rule 0 in the Log Viewer, assuming that anti-spoof logging is enabled on that interface. The remote end of the connection will see a "connection timed out" message.
4. Checks properties and the rulebase.
5. The OS routes the packet. The OS determines that in order to communicate with Client B, it needs to route the packet out the 10.21.31.1 interface.

6. Performs IP Options and anti-spoofing checking on the 10.21.31.1 interface. The destination of the packet (10.21.31.41) is compared against the valid address setting. If the packet is denied because of this check, you will see a reject on Rule 0 in the Log Viewer, assuming that anti-spoofing logging is enabled on that interface. The remote end of the connection will receive a reset, which means a client application will see a "connection refused" message.
7. Checks properties and the rulebase. Properties are always checked outbound as well as inbound. A rules check depends on how you have installed them and how you are enforcing gateway rules.
8. The packet proceeds through the address translation rules. If there was a matching NAT rule, this is where NAT would take place. In this example, NAT is not occurring, so translation is not performed.
9. The packet is sent directly to Client B.

The important detail to note in this process is that NAT is not done until near the very end, that is, after the packet has been routed and has gone through the security policy, but before the packet leaves the gateway. When you do NAT, it means you must make sure that the "untranslated" packet can pass through your anti-spoofing checks and your rulebase.

Implementing NAT, a Step-by-Step Example

The following sample configuration involves NAT. You are shown what you need to do step-by-step to configure FireWall-1 to support this configuration. (See Figure 9-3.)

The security policy is defined as follows:

- Allow the External Mail and Web Server to be reached from anywhere via SMTP and HTTP respectively.
- Allow the External Mail server to send e-mail to anywhere on the Internet and to the Internal Mail Server.
- Allow a second Web instance of a Web Server (running on port 81) to be accessible via a separate IP address on port 80.

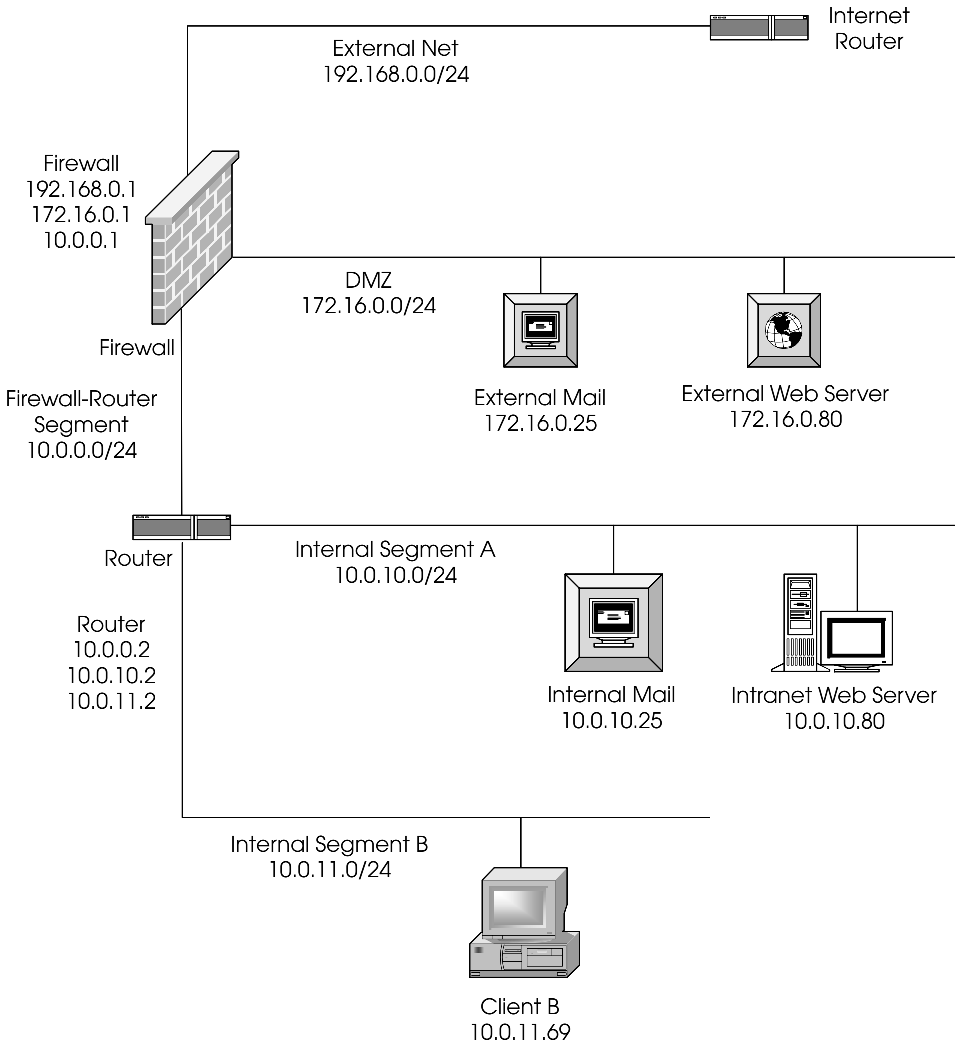


Figure 9-3 Implementing NAT sample network

- Allow clients on Segment A and Segment B to browse the Internet via HTTP or HTTPS hiding behind a single IP address.
- Allow an Intranet Web Server to be accessible on the Internet via HTTP. The Web Server will provide its own authentication, so no authentication is necessary by FireWall-1.
- Except for the former requirements, deny all other traffic.

The preceding policy is specially crafted for demonstration purposes only. Generally speaking, it is not wise to permit traffic from the Internet all the way into the internal network without some sort of encryption. Let's take the following steps to set this up:

- Determine which IP addresses will be used for translation.
- Set up the necessary proxy-ARPs.
- Set up the necessary static host routes.
- Create the necessary network objects.
- Make the necessary modifications to anti-spoofing.
- Create the necessary rulebase rules to permit the desired traffic.
- Create the NAT rules.
- Install the security policy and verify that everything works as planned.

Determining Which IP Addresses Will Be Used

The "legal" addresses include everything in 192.168.0.0/24 except for the firewall (.1) and the router (.2). You can choose any other IP address in the range. The following hosts will use the following static mappings:

- *External Mail Server:* 192.168.0.10
- *Web Server:* 192.168.0.11
- *Web Server (instance on port 81):* 192.168.0.12
- *Intranet Web Server:* 192.168.0.13

For the browsing that Segment A and Segment B hosts will need, use the firewall's external IP address of 192.168.0.2.

Proxy-ARPs

Before you begin, you need to determine which MAC address you are going to use to ARP for the translated IP addresses. You know that all of the translated addresses are on the same subnet as the external interface of the firewall. You simply need to determine what the MAC (or physical) address of the external interface is and use that address. To do this, use the following command:

UNIX and Nokia platform: `ifconfig -a`

Windows NT/2000: `ipconfig /all`

On a UNIX platform, you will see something like this:[5]

```
lo0: flags=849 <UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
        inet 127.0.0.1 netmask ff000000
le0: flags=863 <UP,BROADCAST,NOTRAILERS,RUNNING,
MULTICAST> mtu 1500
        inet 192.168.0.1 netmask ffffff00 broadcast
        192.168.0.255
        ether 0:11:22:33:44:55
le1: flags=863 <UP,BROADCAST,NOTRAILERS,RUNNING,
MULTICAST> mtu 1500
        inet 10.0.0.1 netmask ffffff00 broadcast
        10.0.0.255
        ether 0:c0:78:2:0:d6
le2: flags=863 <UP,BROADCAST,NOTRAILERS,RUNNING,
MULTICAST> mtu 1500
        inet 172.16.0.1 netmask ffffff00 broadcast
        172.16.0.255
        ether 0:c0:78:20:0:6d
```

On a Nokia platform, the output is slightly different:

```
loop0c0:  flags=57<UP,PHYS_AVAIL,LINK_AVAIL,LOOPBACK,
MULTICAST>
        inet mtu 63000 127.0.0.1 —> 127.0.0.1
        phys loop0 flags=10b<UP,LINK,LOOPBACK,PRESENT>
tun0c0:  lname tun0c0
```

[5] On a Solaris platform, it is likely that you will see the same MAC address on all Ethernet interfaces. The default behavior is to use a hostid-based MAC address and not the hardware MAC. As long as two or more interfaces are not on the same physical network, this should not be a problem. You can change the MAC address on a per interface basis with the `ifconfig` command.

```
flags=cf<UP,PHYS_AVAIL,LINK_AVAIL,POINTOPOINT,MULTICAST>
encaps vpn
        phys tun0 flags=107<UP,LINK,POINTOPOINT,PRESENT>
eth-s1p1c0:  lname eth-s1p1c0
flags=e7<UP,PHYS_AVAIL,LINK_AVAIL,BROADCAST,MULTICAST>
        inet mtu 1500 192.168.0.1/24 broadcast
        192.168.0.255
        phys eth-s1p1 flags=133<UP,LINK,BROADCAST,
        MULTICAST,PRESENT>
       ether 0:11:22:33:44:55 speed 100M full duplex
eth-s2p1c0:  lname eth-s2p1c0
flags=e7<UP,PHYS_AVAIL,LINK_AVAIL,BROADCAST,MULTICAST>
        inet mtu 1500 10.0.0.1/24 broadcast 10.0.0.255
        phys eth-s2p1 flags=133<UP,LINK,BROADCAST,
        MULTICAST,PRESENT> ether 0:c0:78:2:0:d6 speed
        100M full duplex
eth-s3p1c0:  lname eth-s3p1c0
flags=e7<UP,PHYS_AVAIL,LINK_AVAIL,BROADCAST,MULTICAST>
        inet mtu 1500 172.16.0.1/24 broadcast
        172.16.0.255
        phys eth-s3p1 flags=133<UP,LINK,BROADCAST,
        MULTICAST,PRESENT> ether 0:c0:78:20:0:6d speed
        100M full duplex
```

On a Windows NT or 2000 platform, you will see this:

```
Ethernet adapter 3C5x91:
        Description . . . . . . . . : 3Com 3C5x9 Ethernet
                                      Adapter
        Physical Address. . . . . . : 00-11-22-33-44-55
        DHCP Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 192.168.0.1
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . : 192.168.0.254
Ethernet adapter 3C5x92:
        Description . . . . . . . . : 3Com 3C5x9 Ethernet
                                      Adapter
        Physical Address. . . . . . : 00-C0-78-20-00-6D
        DHCP Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 10.0.0.1
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . :

Ethernet adapter 3C5x93:
        Description . . . . . . . . : 3Com 3C5x9 Ethernet
                                      Adapter
        Physical Address. . . . . . : 00-0C-87-02-00-D6
        DHCP Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 172.16.0.1
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . :
```

Use the ether or physical address of the system's external interface. In this case, you will use 00:11:22:33:44:55. Now that you know what that MAC address is, you can set up the ARPs. On UNIX systems, this is done as follows:

```
arp -s 192.168.0.10 00:11:22:33:44:55 pub
arp -s 192.168.0.11 00:11:22:33:44:55 pub
arp -s 192.168.0.12 00:11:22:33:44:55 pub
arp -s 192.168.0.13 00:11:22:33:44:55 pub
```

In order for these ARPs to be available on reboot, you need to add them to a file that executes on startup. Do not add them to the /etc/rc3.d/S95firewall1 script, which gets overwritten during an upgrade. Create a new startup script like /etc/rc3.d/S94nat.

Windows NT does not have a proxy-ARP facility, so Check Point has included it as part of the software. Create the file %FWDIR%\state\local.arp and enter the following information:

```
192.168.0.10    00-11-22-33-44-55
192.168.0.11    00-11-22-33-44-55
192.168.0.12    00-11-22-33-44-55
192.168.0.13    00-11-22-33-44-55
```

These ARPs will not become active until a policy reload is performed. In some cases, it may be necessary to stop and start FireWall-1.

NOTE! The Windows 2000 version of FireWall-1 does not support local.arp prior to version 4.1 SP4. Refer to article ID sk699 in Check Point's Knowledge Base for a workaround.

NOTE! FireWall-1's proxy-ARP function will not work in Windows 2000 if you are running Microsoft's Routing and Remote Access Service. Microsoft has produced a hotfix to this issue. To obtain this hotfix, refer to article ID Q82312 in Microsoft's Knowledge Base.

On the Nokia platform running IPSO 3.1 and later, add these ARPs via the Voyager interface as "Proxy-Only" type. In a VRRP configuration, configure both firewalls and use the VRRP MAC address instead of the network card's MAC. You may also configure the NAT IPs as VRRP backup IPs, thus eliminating the need for proxy ARPs.

NOTE! Do not configure the NAT IPs as VRRP backup IPs in FireWall-1 4.1 SP2 because there are some bugs with the anti-spoofing code.

NOTE! In IPSO 3.3 or later, there is an option to allow connections to VRRP IP addresses. Make sure this option is disabled if you plan on configuring the NAT IPs as VRRP IPs.

NOTE! There is a bug with proxy-ARP when used with VRRP MAC addresses that causes some switches to become confused. IPSO 3.3 FCS8, IPSO 3.3.1 FCS7, and IPSO 3.4 and later resolve this problem. Refer to Resolution 3324 in the Nokia Knowledge Base for more details.

Static Host Routes

The only translations for which you need to set up static host routes are those that involve a destination static translation (i.e., where the destination IP address needs to be translated). In this case, you need to set up static host routes for all of them because they will all be connected to by their translated IP address.

You need to determine where the "real" hosts for the virtual IPs are in relation to the firewall. This is so you can determine the "next hop" for the static host routes you will set up. Using Figure 9-3, you know the following information:

- The external mail and external Web servers are on the same subnet as the firewall. In this case, you simply use the "real" host's IP address as the next hop.
- The intranet Web server is not on the same subnet as the firewall. In this case, you want to use the next hop IP address, which is the router that is connected to Segment A—the segment on which the intranet Web server is connected. This is 10.0.0.2.

On UNIX platforms (not IPSO), you would add the static routes like this:[6]

```
route add 192.168.0.10 172.16.0.25 1
route add 192.168.0.11 172.16.0.80 1
route add 192.168.0.12 172.16.0.80 1
route add 192.168.0.13 10.0.0.2 1
```

Like the previous ARPs, these lines need to go into a startup file so that they are available after a reboot. On Windows NT platforms, the static routes are similar:

```
route add 192.168.0.10 172.16.0.25 -p
route add 192.168.0.11 172.16.0.80 -p
route add 192.168.0.12 172.16.0.80 -p
route add 192.168.0.13 10.0.0.2 -p
```

Note that on Windows NT/2000, if you use the `-p` flag, the routes are persistent; that is, they are stored in the Registry and will stay there until they are deleted, even after a reboot.

On a Nokia platform, you add these static routes via Voyager.

[6] There is a way to add static routes on IPSO using the command line. However, you do not use the route command, you issue three separate `dbset` commands. Resolution 1783 in Nokia's Knowledge Base contains a program called addstatic that makes use of these commands.

Network Objects

You must create network objects for both translated and untranslated objects as well per Table 9-2.

Table 9-2 Network objects to create

Name	Object Type	IP/Mask/Group Objects	Description
net-dmz	Network	172.16.0.0/255.255.255.0	Your DMZ
smtp-dmz	Workstation	172.16.0.25	Mail Server in the DMZ
smtp-dmz-ext	Workstation	192.168.0.10	Translated version of smtp-dmz
web-server	Workstation	172.16.0.80	Web Server in the DMZ
web-server-ext	Workstation	192.168.0.11	Translated version of web-server (for port 80)
web-server-ext2	Workstation	192.168.0.12	Translated version of web-server (for port 81)
net-router-segment	Network	10.0.0.0/255.255.255.0	egment shared by firewall S and internal router
net-segment-a	Network	10.0.10.0/255.255.255.0	Segment A
web-intranet	Workstation	10.0.10.80	Intranet Web Server
web-intranet-ext	Workstation	192.168.0.13	Translated version of web-intranet
smtp-internal	Workstation	10.0.10.25	Internal SMTP Server
net-segment-b	Network	10.0.11.0/255.255.255.0	Segment B
valid-dmz	Group	net-dmz + web-server-ext + web-server-ext2 + smtp-server-ext	Represents your DMZ interface's valid addresses for anti-spoofing.
valid-internal	Group	net-segment-a + net-segment-b + web-intranet-ext + net-router-segment	Represents your internal interface's valid addresses for anti-spoofing
Firewall	Workstation	192.168.0.1	Your firewall

Anti-Spoofing

When configuring your firewall object, set your valid address settings according to the settings shown in Table 9-3.

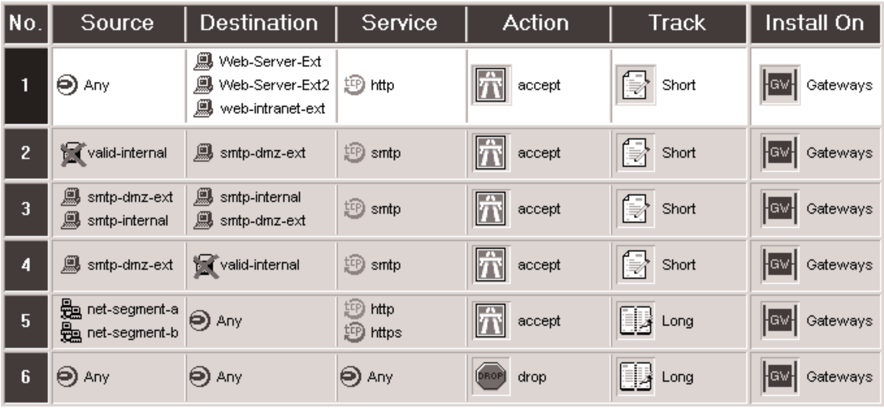
Table 9-3 Valid address settings for firewall

Interface	Valid Address Setting
DMZ	Specific: valid-dmz
Internal	Specific: valid-internal
External	Others

These settings are configured on the Interfaces tab. Also, make sure that each interface has the Spoof Tracking set to "Log" to catch any errors in the anti-spoofing configuration.

Security Policy Rules

The rules created are based on the security policy defined earlier in the "Implementing NAT, a Step-by-Step Example" section. (See Figure 9-4.)

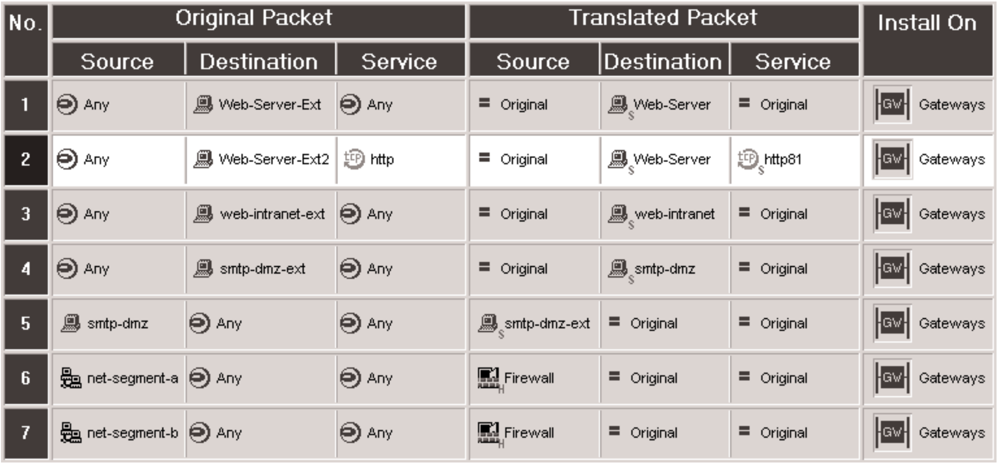


No.	Source	Destination	Service	Action	Track	Install On
1	Any	Web-Server-Ext Web-Server-Ext2 web-intranet-ext	http	accept	Short	Gateways
2	valid-internal	smtp-dmz-ext	smtp	accept	Short	Gateways
3	smtp-dmz-ext smtp-internal	smtp-internal smtp-dmz-ext	smtp	accept	Short	Gateways
4	smtp-dmz-ext	valid-internal	smtp	accept	Short	Gateways
5	net-segment-a net-segment-b	Any	http https	accept	Long	Gateways
6	Any	Any	Any	drop	Long	Gateways

Figure 9-4 Security policy for sample NAT configuration

Address Translation Rules

Once the security policy is defined, NAT rules must be defined. Before you begin, make sure you define a service for port 81. It will be a service of type TCP. In Figure 9-5, it will be referred to as "http81", so you can do the port translation that the security policy requires. Note that the (S) refers to "static" rules and the (H) refers to "hide" rules. (See Figure 9-5.)



No.	Original Packet			Translated Packet			Install On
	Source	Destination	Service	Source	Destination	Service	
1	Any	Web-Server-Ext	Any	= Original	Web-Server S	= Original	GW Gateways
2	Any	Web-Server-Ext2	http	= Original	Web-Server S	http81 S	GW Gateways
3	Any	web-intranet-ext	Any	= Original	web-intranet S	= Original	GW Gateways
4	Any	smtp-dmz-ext	Any	= Original	smtp-dmz S	= Original	GW Gateways
5	smtp-dmz	Any	Any	smtp-dmz-ext S	= Original	= Original	GW Gateways
6	net-segment-a	Any	Any	Firewall H	= Original	= Original	GW Gateways
7	net-segment-b	Any	Any	Firewall H	= Original	= Original	GW Gateways

Figure 9-5 NAT policy for sample NAT configuration

Install the Security Policy and Test

Initiate a connection to exercise each rule to ensure that each rule is functioning as you expect. Test access from inside and outside the network.

Limitations of NAT

NAT does not work in all cases. The following sections document some of the instances where NAT will not work as expected.

The main components that NAT changes are the IP addresses in the TCP/IP headers, and possibly the TCP or UDP ports. This works for some applications, but many applications embed IP addresses in the data portion of the packet (e.g., Microsoft Networking[7]) or expect packets to come from a particular source port (e.g., IKE negotiations for IPSec). In these cases, NAT has to act somewhat like an application proxy in that it must understand the underlying protocol and make intelligent changes to the packets so that the protocol will work despite undergoing NAT.

FireWall-1 understands certain protocols like FTP, RealAudio, and Microsoft Networking (if support is specifically enabled in FireWall-1 4.1 SP1 and later). There are plenty of applications that do not work correctly with FireWall-1's NAT, not necessarily because they are impossible to make work with NAT, but because Check Point has not added support for them. However, there are some protocols that are simply impossible to make work with NAT. Any protocol that uses IP datagram types other than TCP or UDP often fail when NAT is applied. Protocols that validate the IP packet headers between source and destination (such as the Authentication Header mechanism of IPSec) will not work with NAT. To a protocol that protects network traffic from man in the middle attacks—attacks where the headers or payload changes in transit—NAT looks like a hacker. The bottom line: NAT breaks end-to-end connectivity and should only be employed in instances where you can live with the limitations.

NAT will be problematic in situations where the firewall is not between both the source and the destination. Using the example in the previous step-by-step configuration (see Figure 9-3), consider the situation where a host on Segment B (10.0.11.69) tries to access the Intranet Web Server via the translated IP address (192.168.0.13). The host 10.0.11.69 tries to initiate a connection to 192.168.0.13. Routing will eventually take this packet to the firewall. The packet is accepted by the firewall's security policy and is then processed by NAT. The first rule that matches the packet is Rule 3, which translates the destination of the packet from 192.168.0.13 to 10.0.10.80. The "source" of the packet is not changed (the rule says not to touch it). The packet will then be routed back to 10.0.10.80 via 10.0.0.2.

When 10.0.10.80 sends its reply, it is sent to 10.0.11.69 (the "source" of the connection attempt). The reply is routed to 10.0.10.2 and then directly to 10.0.11.69. The host 10.0.11.69 expects replies from 192.168.0.13 (which it tries to connect to), not 10.0.10.80, so the reply packets are ignored.

[7] Microsoft used to say that Microsoft Networking was incompatible with NAT and that NAT, if present in your network, should be removed. It would not surprise me if they still stood by this claim.

What would happen if the rule hides 10.0.11.0/24 behind the firewall's external IP address? When 10.0.11.69 tries to access 10.0.10.80, the packet gets routed to the firewall and passes through the rulebase. NAT then would rewrite the source of the packet to be 192.168.0.2. The destination of the packet would still be 192.168.0.13 (i.e., it does not get translated), but gets routed out the internal interface. The Internet router sees this packet and routes it back to the firewall (it is an external address, after all). The packet would ping-pong back and forth until the packet's Time to Live (TTL) value expired.

One reason you might connect to the translated IP address is because your internal client's DNS server resolves the host's name to the external address. You can resolve this problem by implementing *split-horizon DNS*, that is, maintaining an "internal" version of your DNS and an "external" version of your DNS, typically on separate servers. The external DNS would be accessible from the Internet and contain only a subset of names and addresses contained in the internal DNS server. An internal DNS contains all the names used internally and reflects the internal IP address for a host. The external DNS server reflects the externally resolvable IP addresses for the host.

Other than implementing split-horizon DNS, can you get around this problem? Yes, there are two tricks you can use, which are documented in the following sections. However, it is highly recommended you not place yourself in a position where you have to use these tricks.

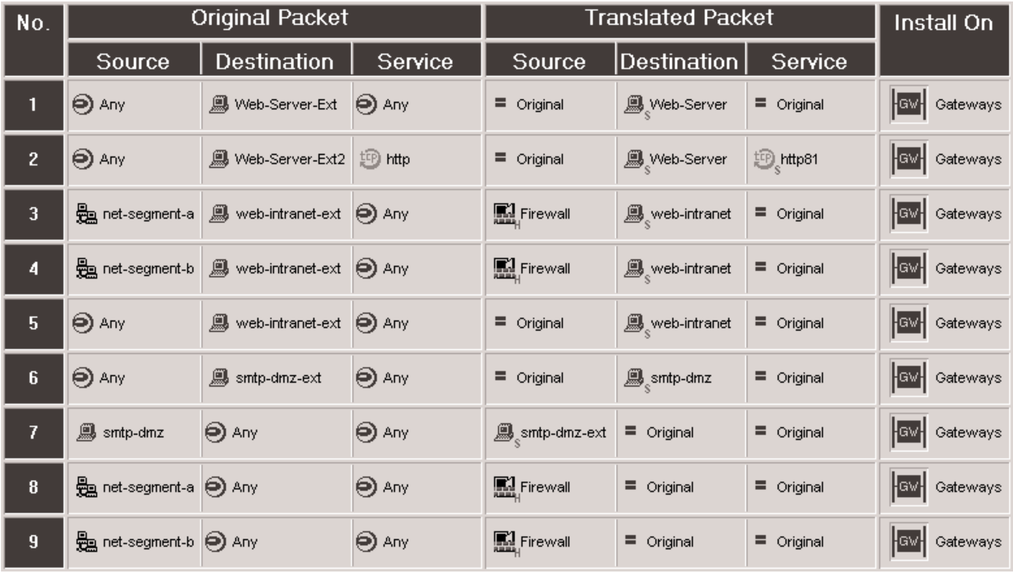
Dual NAT (Translating Both Source and Destination)

FireWall-1 allows you to translate both the source and destination IP address at once. It is simply a matter of crafting the correct rules and placing them in the right order. In the preceding case, if you want to allow your internal network to access the internal host via its translated IP address, modify your NAT rules so they read as shown in Figure 9-6.

The two rules that were added are shown in Figure 9-7.

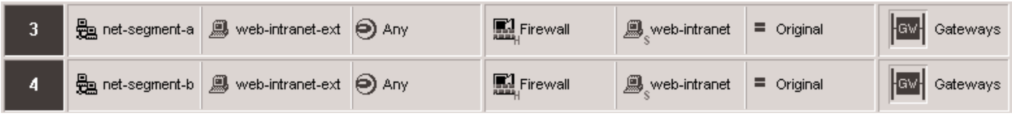
These rules will hide the source address behind the firewall's IP address and modify the destination IP to be the web-intranet address.

In this particular case, there is another issue to contend with: ICMP Redirects. Because the firewall will be routing a packet out the same interface from which it was received, the system sends the client an ICMP Redirect, giving it a more "direct" route to the host. Depending on the exact circumstances, the ICMP Redirect will either cause the connection to never take place or take a long



No.	Original Packet			Translated Packet			Install On
	Source	Destination	Service	Source	Destination	Service	
1	Any	Web-Server-Ext	Any	Original	Web-Server	Original	Gateways
2	Any	Web-Server-Ext2	http	Original	Web-Server	http81	Gateways
3	net-segment-a	web-intranet-ext	Any	Firewall	web-intranet	Original	Gateways
4	net-segment-b	web-intranet-ext	Any	Firewall	web-intranet	Original	Gateways
5	Any	web-intranet-ext	Any	Original	web-intranet	Original	Gateways
6	Any	smtp-dmz-ext	Any	Original	smtp-dmz	Original	Gateways
7	smtp-dmz	Any	Any	smtp-dmz-ext	Original	Original	Gateways
8	net-segment-a	Any	Any	Firewall	Original	Original	Gateways
9	net-segment-b	Any	Any	Firewall	Original	Original	Gateways

Figure 9-6 NAT policy with dual NAT rules



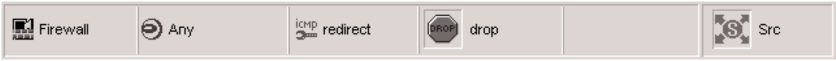
3	net-segment-a	web-intranet-ext	Any	Firewall	web-intranet	Original	Gateways
4	net-segment-b	web-intranet-ext	Any	Firewall	web-intranet	Original	Gateways

Figure 9-7 Dual NAT rules added to Figure 9–6

time to establish as the client will be trying to communicate directly to a host using an IP address it knows nothing about. There are a couple of ways around this situation:

- Bind the "translated" IP address to the server's loopback interface. See the next section for details.
- Block ICMP Redirects. You can block outgoing ICMP Redirects in FireWall-1 with the FireWall-1 rule shown in Figure 9-8.
- On some operating systems, there is an option to disable sending ICMP Redirects. On Solaris, you do this by typing:

```
/usr/sbin/ndd -set /dev/ip ip_send_redirects 0
```



Firewall	Any	icmp redirect	drop		Src

Figure 9-8 Rule to block ICMP Redirects

On a Nokia platform, this can be done on a per interface basis by typing:

```
ipsctl -w interface:<physical-
interface>:family:inet:flags:icmp_no_rdir 1
```

where you replace *<physical-interface>* with the physical interface name (e.g., eth-s1p1).

NOTE! By default, ICMP Redirects are not enabled on any interface running VRRP. This is highly recommended in a VRRP configuration as it limits the possibility that your machine's physical address is propagated.

Assuming this trick works, a side effect can occur, which makes traffic traverse your network twice: once to the firewall and once to the server. This could add to an already congested network.

Binding the NAT IP Address to the Loopback Interface

The basic idea is to bind the translated IP address to the loopback interface of the server. On Windows NT, you need to add the MS Loopback interface (a software-only network adaptor) and add the IP address to this interface with a net mask of 255.255.255.255. In IPSO, you can simply add an IP address to the loop0c0 interface via Voyager. On UNIX machines, use a command such as:

```
ifconfig lo0:0 204.32.38.25 up
```

If packets come into the system for the translated IP address (because, for instance, they did not come to the firewall), the system will respond to packets for this IP address. This method does require slightly more administration because you must also maintain the NAT on the individual servers.

Troubleshooting NAT

To troubleshoot NAT, you should first verify that each necessary step has been performed. This means:

- Validate that an ARP entry exists for the translated IP (or that the translated IP is somehow being routed to the firewall).
- Validate that a static host route exists on the firewall to route the translated IP address to either the untranslated address or the "next hop" address if the real system is more than one hop away from the firewall.
- Validate anti-spoofing. Make sure that the destination IP address is being translated and verify that the translated IP address will pass anti-spoofing checks.
- Validate that the rules are set up correctly. Set any security policy rule that applies to a NAT host to track "long" and ensure that address translation is happening as you expect.

Wherever a verification of the configuration fails, a packet sniffer can be your friend. The remainder of this section shows you what you should see in a packet sniffer, what you shouldn't, and how to fix it.

Although there are plenty of external packet sniffing devices, they can be expensive and inconvenient to use. Fortunately, some operating systems come with their own. Solaris comes with a tool called `snoop`. IPSO, Linux, and AIX come with `tcpdump`. Both of these tools will be discussed briefly in this chapter. Windows NT/2000 machines come with a limited packet sniffer in Network Monitor, but you can obtain a free copy of Ethereal from www.ethereal.com, which works far better.

Since version 4.0, FireWall-1 has also come with its own packet sniffing utility called `fw monitor`. Because it works at the same level as FireWall-1 (i.e., just after the MAC layer and before the network layer), its use in troubleshooting NAT issues is limited. `fw monitor` relies on INSPECT code and is discussed in Chapter 13.

Consider the network and configuration that was used in the earlier step-by-step example (see Figure 9-3). Let's assume that host 192.168.42.69 is attempting to connect to 192.168.0.13, the Intranet Web Server, which really resides at 10.0.10.80.

With a successful connection, a tcpdump of the external interface should show you the following (-i is what you use to specify an interface to listen to):

```
# tcpdump -i eth-s1p1
tcpdump: listening on eth-s1p1
18:51:20.806020 arp who-has 192.168.0.13 tell 192.168.0.2
18:51:20.806020 arp reply 192.168.0.13 is-at
0:11:22:33:44:55
18:51:54.135062 192.168.42.69.1777 > 192.168.0.13.80: S
1184222758:1184222758(0) win 16384 <mss 1460,nop,wscale
0,nop,nop,timestamp[|tcp]> (DF) [tos 0x10]
18:51:54.135062 192.168.0.13.80 > 192.168.42.69.1777: S
1332216451:1332216451(0) ack 1184222759 win 32120 <mss
1460,nop,nop,timestamp 2739310[|tcp]> (DF)
18:51:54.415021 192.168.42.69.1777 > 192.168.0.13.80: .
ack 1 win 17376 <nop,nop,timestamp 11362405 2739310> (DF)
[tos 0x10]
```

If you were to use snoop, you would see the following (-d on snoop allows you to specify an interface to listen to):

```
# snoop -d hme0
Using device /dev/hme (promiscuous mode)
192.168.0.2 -> (broadcast)  ARP C Who is 192.168.0.13 ?
192.168.0.1 -> 192.168.0.2 ARP R 192.168.0.13 is
0:11:22:33:44:55
192.168.42.69 -> 192.168.0.13 HTTP C port=1777
192.168.0.13 -> 192.168.42.69 HTTP R port=1777
192.168.42.69 -> 192.168.0.13 HTTP C port=1777
```

Note that you may not necessarily see the ARP packets, especially if the originator of the packet already has the MAC address in its ARP cache. If you see SYN, SYN/ACK, and ACK packets, the connection should be established.

ARPs

The first part of the communication you should see is the request for MAC addresses via an ARP packet. When everything is working correctly, you will see an exchange like the following on the external interface with tcpdump:

```
18:13:20.806020 arp who-has 192.168.0.13 tell 192.168.0.2
18:13:20.806020 arp reply 192.168.0.13 is-at
0:11:22:33:44:55
```

With snoop, it looks like this:

```
192.168.0.2 -> (broadcast)  ARP C Who is 192.168.0.13 ?
192.168.0.1 -> 192.168.0.2 ARP R 192.168.0.13 is
0:11:22:33:44:55
```

If you do only see the first packet over and over again (e.g., the ARP whois), this means that nobody owns or is proving a proxy-ARP for the translated address. Add a proxy-ARP as described previously. In some cases (especially when Windows NT is the firewall), you may need to add a static host route on the external router.

SYN Packets with No Response

You should then see the SYN packet, which looks something like this with tcpdump:

```
18:13:22.040132 192.168.42.69.1777 > 192.168.0.13.80: S
3911298019:3911298019(0) win 16384 <mss 1460,nop,wscale
0,nop,nop,timestamp[|tcp]> (DF) [tos 0x10]
```

With snoop, it looks like this:

```
192.168.42.69 -> 192.168.0.13 HTTP C port=1777
```

If this packet repeats over and over again, one of four things may be wrong:

- The security policy is dropping the packet. Check your logs for drops.
- The packet is being sent to the wrong MAC address.
- The packet is not being routed properly.
- The packet isn't actually getting translated, thus it is getting ignored.

Verify that the MAC address it is being sent to is correct (in this case, it should be 0:11:22:33:44:55). In tcpdump, you do this with the –e flag, which adds the MAC address to the output. In snoop, the only way to do this is with the –v flag, which unfortunately is extremely verbose.

Also in this example, you are only going to show packets coming from host 192.168.0.13 by means of adding host 192.168.0.13 to the end of your tcpdump or snoop command line. This will only show packets going to or from 192.168.0.13.

```
# tcpdump -e -i eth-s1p1 host 192.168.0.13
tcpdump: listening on eth-s1p1
18:21:49.201680 0:aa:bb:cc:dd:ee 0:55:44:33:22:11 ip 82:
192.168.42.69.2000 > 192.168.0.13.80: S
90360382:90360382(0) win 16384 <mss 1460,nop,wscale
0,nop,nop,timestamp[|tcp]> (DF) [tos 0x10]
18:21:54.240965 0:aa:bb:cc:dd:ee 0:55:44:33:22:11 ip 82:
192.168.42.69.2000 > 192.168.0.13.80: S
90360382:90360382(0) win 16384 <mss 1460,nop,wscale
0,nop,nop,timestamp[|tcp]> (DF) [tos 0x10]
18:22:07.209125 0:aa:bb:cc:dd:ee 0:55:44:33:22:11 ip 82:
192.168.42.69.2000 > 192.168.0.13.80: S
90360382:90360382(0) win 16384 <mss 1460,nop,wscale
0,nop,nop,timestamp[|tcp]> (DF) [tos 0x10]

# snoop –v –d hme0 host 192.168.0.13
Using device /dev/hme (promiscuous mode)
ETHER:  ——- Ether Header ——-
ETHER:
ETHER:  Packet 27 arrived at 16:47:50.83
ETHER:  Packet size = 58 bytes
ETHER:  Destination = 0:55:44:33:22:11,
ETHER:  Source      = 0:aa:bb:cc:dd:ee,
ETHER:  Ethertype = 0800 (IP)
ETHER:
IP:   ——- IP Header ——-
IP:
IP:   Version = 4
IP:   Header length = 20 bytes
```

```
IP:   Type of service = 0x00
IP:         xxx. .... = 0 (precedence)
IP:         ...0 .... = normal delay
IP:         .... 0... = normal throughput
IP:         .... .0.. = normal reliability
IP:   Total length = 44 bytes
IP:   Identification = 47535
IP:   Flags = 0x4
IP:         .1.. .... = do not fragment
IP:         ..0. .... = last fragment
IP:   Fragment offset = 0 bytes
IP:   Time to live = 245 seconds/hops
IP:   Protocol = 6 (TCP)
IP:   Header checksum = f23f
IP:   Source address = 192.168.42.69
IP:   Destination address = 192.168.0.13
IP:   No options
IP:
TCP:  ——- TCP Header ——-
TCP:
TCP:  Source port = 2000
TCP:  Destination port = 80 (HTTP)
TCP:  Sequence number = 90360382
TCP:  Acknowledgement number = 0
TCP:  Data offset = 24 bytes
TCP:  Flags = 0x02
TCP:        ..0. .... = No urgent pointer
TCP:        ...0 .... = No acknowledgement
TCP:        .... 0... = No push
TCP:        .... .0.. = No reset
TCP:        .... ..1. = Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 8760
TCP:  Checksum = 0xda2b
TCP:  Urgent pointer = 0
TCP:  Options: (4 bytes)
TCP:    - Maximum segment size = 1460 bytes
TCP:
```

In the preceding case, MAC 0:aa:bb:cc:dd:ee (which is the MAC of the external router) is trying to send to MAC 0:55:44:33:22:11, which is not the correct MAC. This problem can usually be resolved by flushing the ARP cache on the external router and retrying the test.

If the packet is not being routed properly, you could see a reset (RST) packet (see the section "SYN Followed By RST"), or you could see an ICMP Destination Unreachable packet. Verify that the static host route for 192.168.0.13 is pointing to the next hop address (10.0.0.2 as show in Figure 9-3).

If the packet is not actually being translated, you will see it very clearly in a `tcpdump` or `snoop` on the internal interface:

```
# tcpdump -i eth-s1p2 host 192.168.42.69
tcpdump: listening on eth-s1p2
18:13:22.040132 192.168.42.69.1777 > 192.168.0.13.80: S
3911298019:3911298019(0) win 16384 <mss 1460,nop,wscale
0,nop,nop,timestamp[|tcp]> (DF) [tos 0x10]
18:18:25.040168 192.168.42.69.1777 > 192.168.0.13.80: S
3911298019:3911298019(0) win 16384 <mss 1460,nop,wscale
0,nop,nop,timestamp[|tcp]> (DF) [tos 0x10]
18:40:30.040342 192.168.42.69.1777 > 192.168.0.13.80: S
3911298019:3911298019(0) win 16384 <mss 1460,nop,wscale
0,nop,nop,timestamp[|tcp]> (DF) [tos 0x10]
```

```
# snoop -d hme0
Using device /dev/hme (promiscuous mode)
192.168.42.69 -> 192.168.0.13 HTTP C port=1777
192.168.42.69 -> 192.168.0.13 HTTP C port=1777
192.168.42.69 -> 192.168.0.13 HTTP C port=1777
```

Normally, you should see the translated IP address on the internal interface. If you do not see translated packets, check your NAT rules.

SYN Followed By RST

If the packet that follows the SYN is a RST packet (with `snoop`, you need `–V` to see the TCP flags):

```
# tcpdump -i eth-s1p1 host 192.168.0.13
tcpdump: listening on le0
18:13:22.040132 192.168.42.69.1777 > 192.168.0.13.80: S
3911298019:3911298019(0) win 16384 <mss1460,nop,wscale
0,nop,nop,timestamp[|tcp]> (DF) [tos 0x10]
18:13:22.040132 192.168.0.13.80 > 192.168.42.69.1777: R
0:0(0) ack 3911298020 win 0 [tos 0x10]

# snoop -V -d hme0
Using device /dev/hme (promiscuous mode)
________________________________
192.168.42.69 -> 192.168.0.13 ETHER Type=0800 (IP), size
= 58 bytes
192.168.42.69 -> 192.168.0.13 IP  D=192.168.0.13
S=192.168.42.69 LEN=44, ID=47247
192.168.42.69 -> 192.168.0.13 TCP D=80 S=1777 Syn
Seq=3052932309 Len=0 Win=8760 Options=<mss 1460>
192.168.42.69 -> 192.168.0.13 HTTP C port=1777
________________________________
192.168.0.13 -> 192.168.42.69 ETHER Type=0800 (IP), size
= 60 bytes
192.168.0.13 -> 192.168.42.69 IP  D=192.168.42.69
S=192.168.0.13 LEN=40, ID=61295
192.168.0.13 -> 192.168.42.69 TCP D=64836 S=80 Rst
Ack=3052932310 Win=0
192.168.0.13 -> 192.168.42.69 HTTP R port=1777
```

then one of three things is wrong:

- The firewall is rejecting the packet on anti-spoofing. Verify that 192.168.0.13 is permitted by your anti-spoofing configuration on the internal interface.
- The remote server is not running the service specified (in this case, port 80, HTTP).
- The packet is being routed incorrectly.

If the remote server isn't actually running the service, you will see the following in a `tcpdump` and `snoop` on the internal interface:

```
# tcpdump -i eth-s1p1 host 10.0.10.80
tcpdump: listening on le0
18:13:22.040132 192.168.42.69.1777 > 10.0.10.80.80: S
3911298019:3911298019(0) win 16384 <mss 1460,nop,wscale
0,nop,nop,timestamp[|tcp]> (DF) [tos 0x10]
18:13:22.040132 10.0.10.80.80 > 192.168.42.69.1777: R
0:0(0) ack 3911298020 win 0 [tos 0x10]
```

```
# snoop –V -d hme0
Using device /dev/hme (promiscuous mode)
________________________________
192.168.42.69 -> 10.0.10.80 ETHER Type=0800 (IP), size =
58 bytes
192.168.42.69 -> 10.0.10.80 IP  D=10.0.10.80
S=192.168.42.69 LEN=44, ID=47247
192.168.42.69 -> 10.0.10.80 TCP D=80 S=1777 Syn
Seq=3052932309 Len=0 Win=8760 Options=<mss 1460>
192.168.42.69 -> 10.0.10.80 HTTP C port=1777
________________________________
192.168.0.13 -> 192.168.42.69 ETHER Type=0800 (IP), size
= 60 bytes
192.168.0.13 -> 192.168.42.69 IP  D=192.168.42.69
S=10.0.10.80 LEN=40, ID=61295
192.168.0.13 -> 192.168.42.69 TCP D=64836 S=80 Rst
Ack=3052932310 Win=0
192.168.0.13 -> 192.168.42.69 HTTP R port=1777
```

The internal interface should see the untranslated packets (10.0.10.80 is the system's real IP). If the packet is being routed to the wrong interface, you will also see the same behavior as in the preceding output, but you will see a reject on Rule 0 in the Log Viewer as well. Verify that the static host route is set up correctly.

Useful tcpdump Flags

Table 9-4 contains a list of some useful flags for `tcpdump`, which takes commands in the following format:

```
tcpdump –i interface-name [other-flags] [expression]
```

Table 9-4 `tcpdump` Flags

Flag	Description
`-e`	Displays MAC addresses with each packet.
`-i interface`	Required. Specify an interface to listen on.
`-l`	Buffer stdout, which is useful for piping `tcpdump` output to other programs.
`-n`	Disables name resolution on packets shown.
`-p`	Do not put interface in promiscuous mode; i.e., only show packets destined for the host.
`-r filename`	Read `tcpdump` capture from specified file.
`-s N`	Capture N number of bytes for each packet. The default is 68. (Useful with –x or –X.)
`-S`	Print absolute TCP sequence numbers instead of relative ones.
`-w filename`	Write captured packets to specified file.
`-x`	Hex dump of received packets.
`-X`	Hex and ASCII dump of received packets (IPSO only).

tcpdump Expressions

All `tcpdump` commands can be followed by an expression that filters the displayed (or saved) packets so that only the packets that are interesting are shown. Some useful expressions are shown in Table 9-5.

Table 9-5 `tcpdump` Expressions

Expression	Description
port 80	Show all packets with source or destination port 80 (TCP or UDP).
host 192.168.0.3	Show all packets coming from or going to host 192.168.0.3.
host 192.168.0.3 and tcp port 80	Show all packets coming from or going to host 192.168.0.3 and that are TCP packets with a source or destination port of 80.
proto vrrp	Show all VRRP packets. On non-IPSO platforms, use “ip proto 112”.
icmp	Show all ICMP packets.

Expression	Description
\(src host 192.168.0.1 or src host 192.168.0.2\) and proto vrrp	Show all VRRP packets that originate from 192.168.0.1 and 192.168.0.2 (the \ before the parenthesis is to escape it for the shell).
ether host aa:bb: cc:dd:ee:ff	Show all packets that come from or go to the specified MAC address.
ip proto 50	Show all packets of IP Proto 50 (IPSec AH in this example).
ip[2:2] > 576	Show IP packets that are longer than 576 bytes ([2:2] refers to the specific byte location in the TCP header and its length).
tcp[13] & 0x12 != 0	Show only TCP SYN/ACK packets. tcp[13] refers to the 13th byte in the TCP header of the packet.
icmp[0]	Show ICMP type 0 packets (i.e., echo reply). Icmp[0] refers to the 0th byte in the ICMP header.
icmp[0] = 3 and icmp[1] = 4	Show ICMP type 3, code 4 packets. These happen to be a response to receiving a packet that is too large to process and has the Don't Fragment bit set.

Useful snoop Flags

Table 9-6 contains a list of some useful flags for `snoop`, which takes commands in the following format:

```
snoop [flags] [expression]
```

Table 9-6 `snoop` Flags

Flag	Description
`-d interface`	Specify an interface to listen on.
`-v`	Verbose mode. All packet data is displayed for each packet.
`-V`	A less-verbose verbose mode. A summary line is displayed for each layer in the ISO model.
`-n`	Disables name resolution on packets shown.
`-P`	Do not put interface in promiscuous mode, which means only show packets actually destined for this host (not useful on a switched segment).

(*continued*)

Table 9-6 snoop Flags (*cont.*)

Flag	Description
`-i filename`	Read packets previously captured from file filename.
`-o filename`	Write packets to capture file named filename.
`-s numbytes`	Capture numbytes bytes for each packet. Normally, all bytes in the packet are captured.
`-p x,y`	Show packets numbered between x and y. The first packet captured is 1.
`-t [a|d|r]`	Timestamp format: a (absolute, i.e., wall clock time), d (delta, since capture was started), and r (relative time).

snoop Expressions

All snoop commands can be followed by an expression that filters the displayed (or saved) packets so that only the packets that are interesting are shown. Some useful expressions are shown in Table 9-7. Note that many of the expressions are similar to tcpdump.

Table 9-7 snoop Expressions

Expression	Description
port 80	Show all packets with source or destination port 80 (TCP or UDP).
host 192.168.0.3	Show all packets coming from or going to host 192.168.0.3. You can also omit the "host" qualifier as well.
host 192.168.0.3 and tcp port 80	Show all packets coming from or going to host 192.168.0.3 and that are TCP packets with a source or destination port of 80.
Icmp	Show all ICMP packets.
from 192.168.0.1 or to 192.168.0.2	Show all packets that originate from 192.168.0.1 or are destined for 192.168.0.2.
ether aa:bb:cc:dd:ee:ff	Show all packets that come from or go to the specified MAC address.
ip proto 50	Show all packets of IP Proto 50 (IPSec AH in this example).
greater 576	Show packets longer than 576 bytes. You can use the word "less" instead of greater to show packets smaller than 576 bytes.
tcp[13] & 0x12 != 0	Show only TCP SYN/ACK packets. tcp[13] refers to the 13th byte in the TCP header of the packet.

Summary

NAT is necessary because organizations and individuals need the ability to allow more hosts to communicate with the Internet than their address space allows. Specific blocks of IP addresses have been set aside to accommodate NAT.

NAT also provides a way to more efficiently use address space without the overhead imposed by subnetting. However, NAT imposes restrictions of its own, and it may not be appropriate for every situation.

Sample Configurations

The following three situations presented are representative of situations I have come across in the real world. Each is designed to demonstrate what people typically do with NAT and how the situations would be implemented on the chosen platform.

A Simple Network with NAT

The Situation

You work for a small company with a few hosts on a flat network segment. Your firewall runs on a Windows NT platform. The ISP has only given you a /29 net block, which effectively gives you six hosts you can use on the outside segment. Because the firewall and Internet router each need a unique IP address, this leaves a total of four addresses that can be used for other hosts. (See Figure 9-9.)

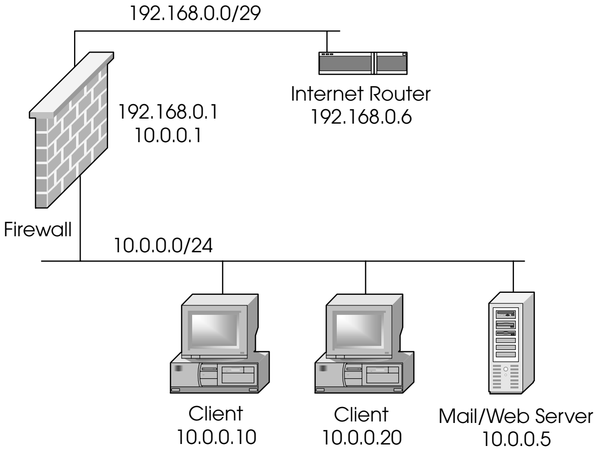


Figure 9-9 Sample configuration network #1

The Goals

- Allow Internet users access to the Mail and Web Servers via SMTP and HTTP, respectively. In the future, these services will be provided on separate systems, so setting up each "service" with a unique IP is desirable to make future migration easier.
- Allow internal users to access anything on the Internet. All outbound users will be hidden behind a single IP address; however, this IP address should be different from the firewall.

Checklist

- Determine which IP addresses will be used for translation.
- Set up the necessary proxy-ARPs.
- Set up the necessary static host routes.
- Create the necessary network objects.
- Make the necessary modifications to anti-spoofing.
- Create the necessary rulebase rules to permit the desired traffic.
- Create the NAT rules.
- Install the security policy.

Implementation

You must first determine which IPs you will use. Your usable IPs are 192.168.0.2 - 192.168.0.5. Let's make 192.168.0.2 the IP you use for your external clients to hide behind, 192.168.0.3 for the SMTP server, and 192.168.0.4 for the HTTP server.

Next, set up the static ARPs for the translated addresses. In order to do this, you need to determine the MAC address of your external interface. Use the command `ipconfig /all` to determine this address:

```
Ethernet adapter 3C5x91 :
        Description . . . . . . . . : 3Com 3C5x9 Ethernet
Adapter
        Physical Address. . . . . . : 00-22-44-66-88-AA
        DHCP Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 192.168.0.1
        Subnet Mask . . . . . . . . : 255.255.255.248
        Default Gateway . . . . . . : 192.168.0.6
Ethernet adapter 3C5x92:
        Description . . . . . . . . : 3Com 3C5x9 Ethernet
Adapter
        Physical Address. . . . . . : 00-00-87-20-66-69
        DHCP Enabled. . . . . . . . : No
        IP Address. . . . . . . . . : 10.0.0.1
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . :
```

The external MAC is `00-22-44-66-88-AA`, which you will enter along with the IP addresses in the `%FWDIR%\state\local.arp`:

```
192.168.0.2 00-22-44-66-88-AA
192.168.0.3 00-22-44-66-88-AA
192.168.0.4 00-22-44-66-88-AA
```

Both 192.168.0.3 and 192.168.0.4 each need a static host route. The 192.168.0.2 address does not need one as users should never be directly connecting to 192.168.0.2. Because the real host is on the same subnet as the firewall, the static route should be directed at the host itself:

```
route -p add 192.168.0.3 10.0.0.5
route -p add 192.168.0.4 10.0.0.5
```

Because you are using the `-p` option, these routes will be available after a reboot; they will be stored in the Registry.

Table 9-8 lists the network objects you will create.

When configuring your firewall object, set your valid address settings according to the settings shown in Table 9-9.

The valid address settings are set on the Interfaces tab. Also, make sure that each interface has the Spoof Tracking set to “Log” to catch any errors in the anti-spoofing configuration.

The rulebase should look similar to the rules shown in Figure 9-10.

The NAT rules should look like the rules shown in Figure 9-11.

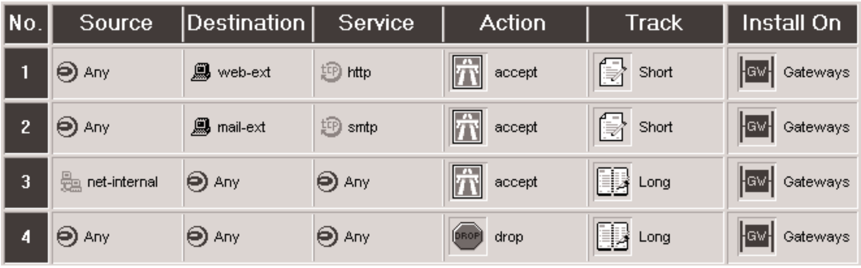
Save and install the policy.

Table 9-8 Network objects for sample configuration #1

Name	Object Type	IP/Net Mask/Group Objects	Description
Net-internal	Network	10.0.0.0/255.255.255.0	The network that represents the Internal Network
mail-web-server	Workstation	10.0.0.5	Mail/Web Server on the Internal Network
mail-ext	Workstation	192.168.0.3	Translated IP for Mail Server
web-ext	Workstation	192.168.0.4	Translated IP for Web Server
external-hide	Workstation	192.168.0.2	IP that users will hide behind when going out
valid-internal	Group	net-internal + mail-ext + web-ext	Represents your internal interface's valid addresses for anti-spoofing
firewall	Workstation	192.168.0.1	Your firewall

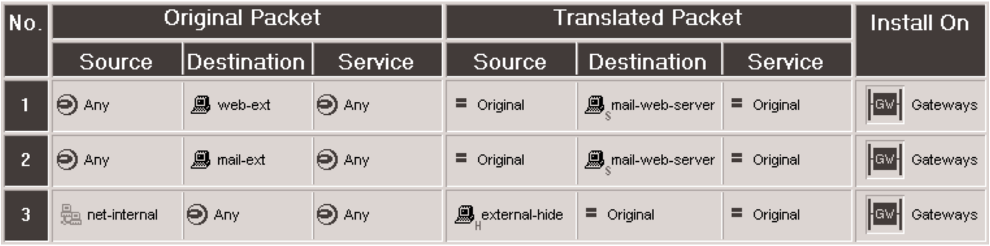
Table 9-9 Valid address settings for firewall

Interface	Valid Address Setting
Internal	Specific: valid-internal
External	Others



No.	Source	Destination	Service	Action	Track	Install On
1	Any	web-ext	http	accept	Short	Gateways
2	Any	mail-ext	smtp	accept	Short	Gateways
3	net-internal	Any	Any	accept	Long	Gateways
4	Any	Any	Any	drop	Long	Gateways

Figure 9-10 Security policy for sample configuration #1



No.	Original Packet			Translated Packet			Install On
	Source	Destination	Service	Source	Destination	Service	
1	Any	web-ext	Any	= Original	mail-web-server	= Original	Gateways
2	Any	mail-ext	Any	= Original	mail-web-server	= Original	Gateways
3	net-internal	Any	Any	external-hide	= Original	= Original	Gateways

Figure 9-11 NAT policy for sample configuration #1

Notes

It is not a wise idea to have your internal hosts on the same LAN segment as hosts that are accessible from an untrusted network like the Internet. However, this is a situation that, for various reasons, all too many security administrators find themselves in. From a security standpoint, you are much better off trying to move externally accessible servers to a DMZ. It will cost a couple of hundred dollars to purchase an extra LAN adaptor, an extra switch or hub, and a few extra cables, but the extra security gained will be well worth it.

You will not be able to access the Mail/Web Server from the internal segment via its translated addresses without some additional configuration.

Migrating to a Larger Net with NAT

The Situation

The company you work for has grown. Your ISP has given you an external segment with a few more addresses (192.168.1.64/28), and you have a separate LAN segment for your DMZ, which now also has a few more hosts in it. Your firewall platform has also changed from Windows NT to Solaris.

A certain amount of "backward compatibility" needs to be maintained with the old setup; that is, certain hosts need to be reachable by their old addresses. For the external addresses, the ISP is continuing to route the 192.168.0.0/29 segment to you until such time as the address space is no longer needed. (See Figure 9-12.)

The Goals

- Allow Internet users access to the Mail and Web Servers via SMTP and HTTP, respectively. Note that the servers will have to be accessible by their old, historical IPs as well as their new IPs.

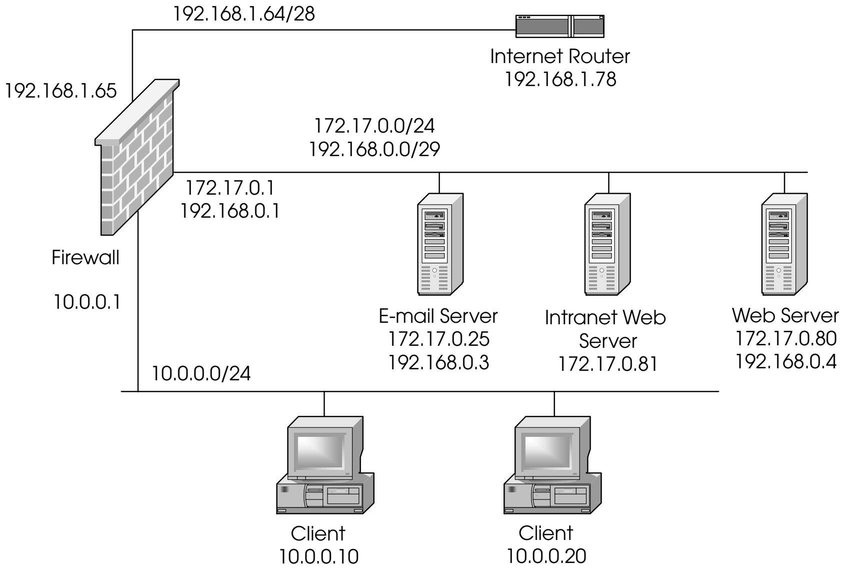


Figure 9-12 Sample configuration network #2

- Allow internal users to access anything on the Internet. All outbound users will be hidden behind a single IP address; however, this IP address should be different from the firewall.
- Allow internal users to access the Intranet Web Server by its old IP address (10.0.0.5). This server will not be accessible from the Internet.

Checklist

- Determine which IP addresses will be used for translation.
- Set up the necessary proxy-ARPs.
- Set up the necessary static host routes.
- Create the necessary network objects.
- Make the necessary modifications to anti-spoofing.
- Create the necessary rulebase rules to permit the desired traffic.
- Create the NAT rules.
- Install the security policy.

Implementation

To simplify your NAT configuration a bit, assign the 192.168.0.0/29 network to the DMZ. Make sure the external router is configured to route all requests for this network to the firewall. You also need to give the firewall an IP address of 192.168.0.1 on the DMZ interface. In Solaris, you add an `/etc/hostname.qe3:1` file with this IP address. You also have to modify `/etc/netmasks` so that 192.168.0.0 has the correct net mask (255.255.255.248). So that you don't have to reboot for this to take effect, execute the following set of commands:

```
# ifconfig qe3:1 plumb
# ifconfig qe3:1 inet 192.168.0.1 netmask 255.255.255.248
broadcast 192.168.0.7 up
```

The SMTP and HTTP servers need to have secondary IP addresses of 192.168.0.3 and 192.168.0.4, respectively. Similar steps need to be taken on these servers.

You must then determine which IPs you will use for translation. Your new usable address range is 192.168.1.66 - 192.168.1.77. Let's make 192.168.1.66 the IP you use for your external clients to hide behind, 192.168.1.67 the new IP for your SMTP server, and 192.168.1.68 for your HTTP server. You are also translating 10.0.0.5 to 172.17.0.81.

Next, set up the static ARPs for the translated addresses. Because you are translating both internal and external addresses to the DMZ, you need both the external and internal interfaces' MAC address. An `ifconfig -a` shows you the following:

```
lo0: flags=849 <UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
        inet 127.0.0.1 netmask ff000000
le0: flags=863 <UP,BROADCAST,NOTRAILERS,RUNNING,
MULTICAST> mtu 1500
        inet 192.168.1.65 netmask fffffff0 broadcast
        192.168.1.79
        ether 0:12:34:56:78:9a
qe0: flags=863 <UP,BROADCAST,NOTRAILERS,RUNNING,
MULTICAST> mtu 1500
        inet 10.0.0.1 netmask ffffff00 broadcast
        10.0.0.255
        ether 8:0:20:6d:0:20
```

```
qe3: flags=863 <UP,BROADCAST,NOTRAILERS,RUNNING,MULTI-
CAST> mtu 1500
        inet 172.17.0.1 netmask ffffff00 broadcast
        172.17.0.255
        ether 8:0:20:20:0:6d
qe3:1 flags=863 <UP,BROADCAST,NOTRAILERS,RUNNING,
MULTICAST> mtu 1500
        inet 192.168.0.1 netmask fffffff8 broadcast
        192.168.0.7
```

The external MAC is 00:12:34:56:78:9a, and the internal MAC address is 8:0:20:20:0:6d. The ARPs you would do are as follows:

```
arp -s 192.168.1.66 0:12:34:56:78:9a pub
arp -s 192.168.1.67 0:12:34:56:78:9a pub
arp -s 192.168.1.68 0:12:34:56:78:9a pub
arp -s 10.0.0.5     8:0:20:20:0:6d   pub
```

Static routes are as follows (note that you still need static routes for the "old" addresses, even if you don't need ARPs for them):

```
route add 192.168.1.67 172.17.0.25 1
route add 192.168.1.68 172.17.0.80 1
route add 10.0.0.5     172.17.0.81 1
```

Because this is a UNIX platform, these ARPs and routes will disappear after a reboot. You need to add these routes and ARPs to a startup file. It is recommended that you create a new script for this purpose (such as `/etc/rc3.d/S94addroutes`), and add the preceding commands to this file.

Table 9-10 shows the network objects that will be created.

When configuring your firewall object, set your valid address settings according to the settings shown in Table 9-11.

These settings are configured on the Interfaces tab. Also, make sure that each interface has the Spoof Tracking set to "Log" to catch any errors in the anti-spoofing configuration.

The rulebase should look similar to the rules shown in Figure 9-13.

The NAT rules should look like the rules shown in Figure 9-14.

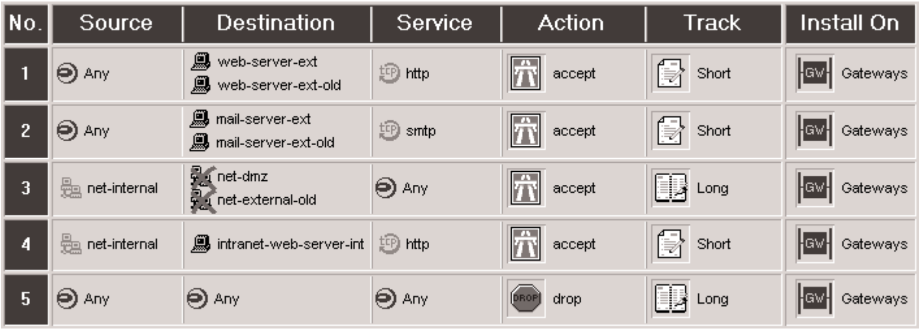
Save and install the policy.

Table 9-10 Network objects for sample configuration

Name	Object Type	IP/Net Mask/Group Objects	Description
net-internal	Network	10.0.0.0/255.255.255.0	The network that represents the Internal Network
net-dmz	Network	172.17.0.0/255.255.255.0	The network that represents the DMZ
net-external-old	Network	192.168.0.0/255.255.255.248	The old external network now on the DMZ
mail-server	Workstation	172.17.0.25	The Mail Server
mail-server-ext	Workstation	192.168.1.67	Translated IP for the Mail Server
mail-server-ext-old	Workstation	192.168.0.3	Translated IP for the Mail Server (historical)
web-server	Workstation	172.17.0.80	The Web Server
web-server-ext	Workstation	192.168.1.68	Translated IP for the Web Server
web-server-ext-old	Workstation	192.168.0.4	Translated IP for the Web Server (historical)
intranet-web-server	Workstation	172.17.0.81	The Intranet Web Server
intranet-web-server-int	Workstation	10.0.0.5	Translated IP for the Intranet Web Server
external-hide	Workstation	192.168.1.66	IP that users will hide behind when going out
valid-dmz	Group	net-dmz + intranet-web-server-int + mail-server-ext + web-server-ext + net-external-old	Represents your DMZ interface's valid addresses for anti-spoofing
valid-internal	Group	net-internal	Represents your internal interface's valid addresses for anti-spoofing
firewall	Workstation	192.168.1.65	Your firewall

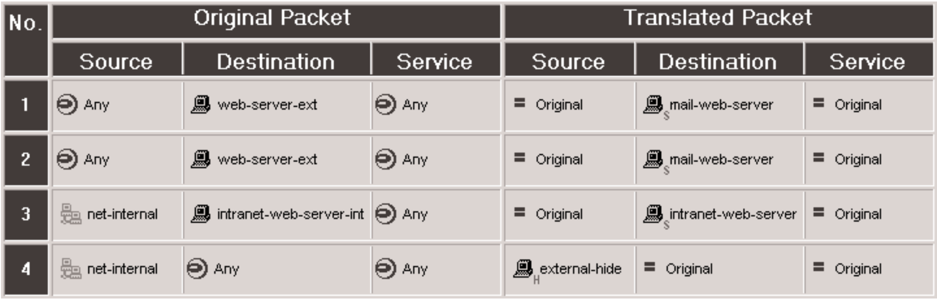
Table 9-11 Valid address settings for firewall

Interface	Valid Address Setting
DMZ	Specific: valid-dmz
Internal	Specific: valid-internal
External	Others



No.	Source	Destination	Service	Action	Track	Install On
1	Any	web-server-ext web-server-ext-old	http	accept	Short	Gateways
2	Any	mail-server-ext mail-server-ext-old	smtp	accept	Short	Gateways
3	net-internal	net-dmz net-external-old	Any	accept	Long	Gateways
4	net-internal	intranet-web-server-int	http	accept	Short	Gateways
5	Any	Any	Any	drop	Long	Gateways

Figure 9-13 Security policy for sample configuration #2



No.	Original Packet			Translated Packet		
	Source	Destination	Service	Source	Destination	Service
1	Any	web-server-ext	Any	Original	mail-web-server	Original
2	Any	web-server-ext	Any	Original	mail-web-server	Original
3	net-internal	intranet-web-server-int	Any	Original	intranet-web-server	Original
4	net-internal	Any	Any	external-hide	Original	Original

Figure 9-14 NAT policy for sample configuration #3

Notes

Sites on the DMZ should be accessible by their translated IP addresses, even from the internal network. This is because the communication is now mediated by the firewall. In the previous example, this was not the case.

Double-Blind Network Configuration

The Situation

There is a device within your network that has a faulty IP implementation and can only talk to hosts on the same subnet as it is on (i.e., it has no concept of routing).[7] Because it is also not desirable to allow everyone to access this host, a firewall is necessary to restrict access to this host. The host is using nonroutable addresses and cannot be seen by the rest of the network. It must be given a "routable" address so that it can be accessed. Because neither side of the connection can know the "true" IP address of its peer, this is referred to as a *double-blind* network configuration.

A Nokia IP330 will be used to protect this device, which will be directly connected to the device via a cross-over cable. The rest of the network (the entire 10.0.0.0/8) is used internally. Figure 9-15 only shows the relevant parts of the network.

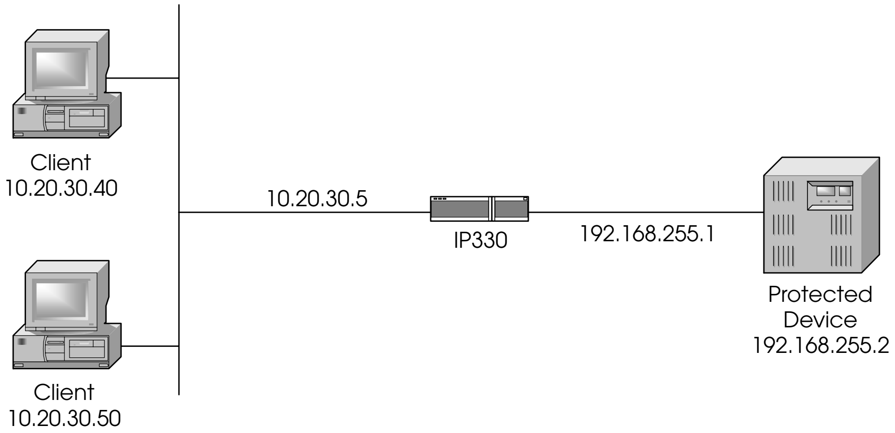


Figure 9-15 Sample configuration network #3

The Goals

- Allow FTP and Telnet access to 10.20.30.6 (the translated IP address for this device).

[7] Don't laugh, I've actually run into this situation with an IP enabled PBX system.

- Allow HTTP access to the IP330 for management purposes from a specific management console (10.250.0.5, not pictured here).
- Allow SSH access to the IP330 for management purposes from anywhere.

Checklist

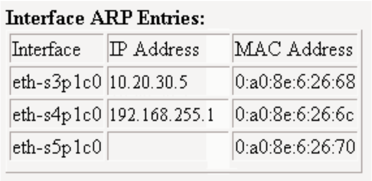
- Determine which IP addresses will be used for translation.
- Set up the necessary proxy-ARPs.
- Set up the necessary static host routes.
- Create the necessary network objects.
- Make the necessary modifications to anti-spoofing.
- Create the necessary rulebase rules to permit the desired traffic.
- Create the NAT rules.
- Install the security policy.

Implementation

From the preceding goals, you know that you will be translating 10.20.30.6 to 192.168.255.2. You also know that all access to this protected device must appear to be coming from a device on the same subnet. The firewall is appropriate in this case.

You need to create an ARP for 10.20.30.6 using the MAC address of the external interface of the IP330. You can easily do this in Voyager. Figure 9-16 shows the interface ARP entries, which are on the ARP configuration page.

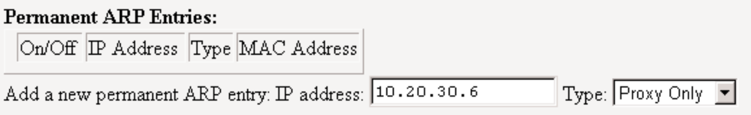
You can see the two interfaces and their MAC addresses. For 10.20.30.6, you will use 0:a0:8e:6:26:68. Right above the Interface ARP entries on the Voyager page is where you create this ARP entry (see Figure 9-17).



Interface ARP Entries:

Interface	IP Address	MAC Address
eth-s3p1c0	10.20.30.5	0:a0:8e:6:26:68
eth-s4p1c0	192.168.255.1	0:a0:8e:6:26:6c
eth-s5p1c0		0:a0:8e:6:26:70

Figure 9-16 Interface ARP entries in Voyager



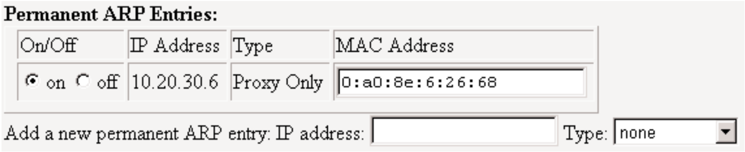
Permanent ARP Entries:

On/Off	IP Address	Type	MAC Address

Add a new permanent ARP entry: IP address: 10.20.30.6 Type: Proxy Only

Figure 9-17 Create ARP entries in Voyager

Click on the Apply button at the bottom of the page. You then need to set the MAC address for this entry (see Figure 9-18).



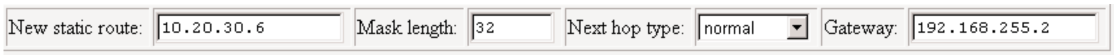
Permanent ARP Entries:

On/Off	IP Address	Type	MAC Address
● on ○ off	10.20.30.6	Proxy Only	0:a0:8e:6:26:68

Add a new permanent ARP entry: IP address: Type: none

Figure 9-18 Add MAC to ARP entries in Voyager

Click on Apply again. Go to the bottom of the static route page in Voyager, and add a static route to route 10.20.30.6 to 192.168.255.2 (see Figure 9-19).



New static route: 10.20.30.6 Mask length: 32 Next hop type: normal Gateway: 192.168.255.2

Figure 9-19 Add static route in Voyager

Click on Apply, and then click on Save. This configuration is now active across reboots.

The network objects you will create are shown in Table 9-12.

Table 9-12 Network objects for sample configuration

Name	Object Type	IP/Net Mask/Group Objects	Description
net-protected-device	Network	192.168.255.0/255.255.255.0	The network that the protected device is on
net-internal	Network	10.0.0.0/255.0.0.0	The internal network
protected-device	Workstation	192.168.255.2	The protected device's internal IP
protected-device-xlate	Workstation	10.20.30.6	Translated IP for the protected device
management-device	Workstation	10.250.0.5	Host allowed to connect to IP330 via HTTP
firewall-eth -s4p1	Workstation	192.168.255.1	A workstation object (defined without FireWall-1 installed) that represents the system's interface facing the protected device. You will need this later.
valid-eth-s4p1	Group	net-protected-device + protected-device-xlate	Represents the valid address setting for eth-s4p1, the interface that the protected device is hooked to
Firewall	Workstation	10.20.30.5	Your firewall

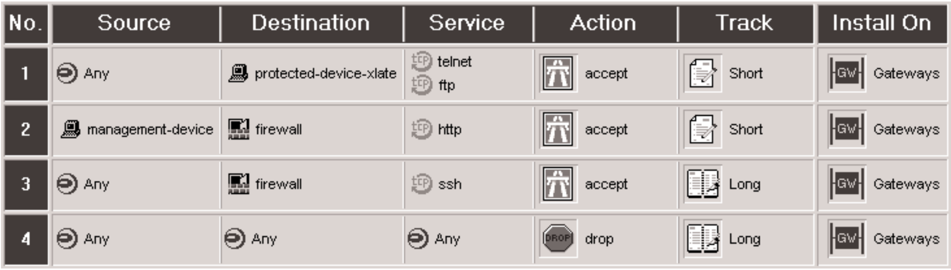
When configuring your firewall object, set your valid address settings according to the settings shown in Table 9-13.

Table 9-13 Valid address setting for firewall

Interface	Valid Address Setting
eth-s4p1c0	Specific: valid-eth-s4p1
eth-s3p1c0	Others

These settings are configured on the Interfaces tab. Also, make sure that each interface has the Spoof Tracking set to "Log" to catch any errors in the anti-spoofing configuration.

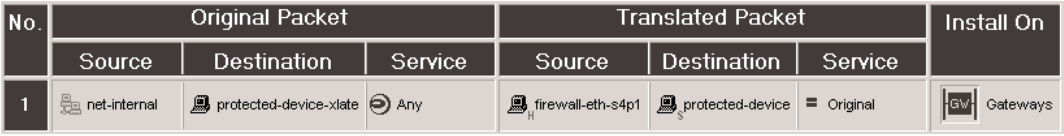
The rulebase should look similar to the rules shown in Figure 9-20.



No.	Source	Destination	Service	Action	Track	Install On
1	Any	protected-device-xlate	telnet ftp	accept	Short	Gateways
2	management-device	firewall	http	accept	Short	Gateways
3	Any	firewall	ssh	accept	Long	Gateways
4	Any	Any	Any	drop	Long	Gateways

Figure 9-20 Security policy for sample configuration #3

The NAT rules should look like the rule shown in Figure 9-21.



No.	Original Packet			Translated Packet			Install On
	Source	Destination	Service	Source	Destination	Service	
1	net-internal	protected-device-xlate	Any	firewall-eth-s4p1	protected-device	Original	Gateways

Figure 9-21 NAT policy for sample configuration #3

Save and install the policy.

Notes

You could do this in the reverse direction as well (i.e., have the protected device access hosts on the other side of the firewall as if they were on the same subnet), but this lab only shows the connections occurring in one direction.