Setting Up a PIX Firewall

Terms you'll need to understand:

- ✓ Privileged mode
- ✓ Unprivileged mode
- ✓ Configuration mode
- ✓ Monitor mode
- ✓ **show xlate**
- ✓ **clear xlate**
- ✓ **passwd**
- ✓ **reload**
- ✓ **write**
- ✓ **show version**

Techniques you'll need to master:

- ✓ Access modes
- ✓ Default interface names
- ✓ Setting a Telnet password
- ✓ Copying an image from a TFTP server
- ✓ Password recovery

Cisco's Command Line Interface (CLI) is the main tool used to configure the PIX firewall. The CLI is a text-based interface you can connect to using the console port or Telnet. The PIX does have a Web interface, called the PIX Device Manager, which is discussed further in Chapter 13, "IPSec and Virtual Private Networks." This chapter covers some important commands needed to execute, monitor, and back up your PIX configurations.

Factory Default Configurations

The default configuration of the PIX is dependent on the model of firewall you have. The PIX 501 and 506E both come with basic configurations designed for SOHO and ROBO environments. The PIX 515E, 525, and 535 have no basic configuration settings; apparently Cisco figures that if you buy an expensive firewall then you had better know how to configure it!

Cisco PIX 501 and 506E Default Settings

Both models are ready right out of the box, with the following configurations set to their default settings. Tables 4.1 and 4.2 display the inside and outside default configuration settings.

Table 4.1 Ethernet 0 (Outside)

Configuration	Setting
Interface name	outside
Security level	0
DHCP	client

Table 4.2 Ethernet 1 (Inside)

Configuration	Setting
Interface name	inside
Security level	100
IP address/subnet mask	192.168.1.1 255.255.255.0
DHCP	Allows clients to automatically obtain an IP address from the PIX

The traffic flow is set to the default, which means that traffic is allowed to travel from the inside (100) to the outside (0) normally. Any traffic from the outside (0) to the inside (100) is not allowed, however. The enable password is blank, and the Telnet password is `cisco`, all lowercase.

CLI Administrative Access Modes

The CLI has several administrative access modes that are similar to other Cisco equipment. Similarly, the commands you're allowed to execute are defined by the current access mode. Unprivileged, privileged, configuration, and monitor are the access modes covered in this section.

Unprivileged Mode

Unprivileged mode, also known as user EXEC mode, contains a > symbol at the prompt. This is the first access mode you come to when entering the CLI, and it allows only a very small subset of the available commands. The question mark command displays the available commands in unprivileged mode:

```
pixfirewall> ?
enable     Turn on privileged commands
help      Help list
login      Log in as a particular user
logout     Exit from current user profile, and to unprivileged mode
pager      Control page length for pagination
quit      Quit from the current mode, end configuration or logout
pixfirewall>
```

The commands in unprivileged mode can't actually change any configuration settings, but they do allow you to move to the next level—privileged EXEC mode.

Privileged Mode

Privileged mode, also known as privileged EXEC mode, is symbolized by a pound sign (#) at the prompt. Privileged EXEC mode gives you the full set of available commands that enable you to configure your PIX firewall. To enter this mode, you need to type the word `enable` and enter the password at the user EXEC prompt. To move back to user EXEC mode, you must type the command `disable`. Listing 4.1 shows how to use the `enable` and `disable` commands to enter and exit privileged EXEC mode.

Listing 4.1 The enable and disable Commands

```
pixfirewall> enable
Password:
pixfirewall# disable
pixfirewall>
```

Configuration Mode

Configuration mode is represented by a `(config)#` prompt. This mode allows access to interfaces, virtual private networks (VPNs), DCHP servers, hostname settings, and so on. You can enter this mode by entering the command `config terminal` at the privileged EXEC prompt. To return to privileged EXEC mode, you must type `exit` (or `disable` to return even further back to unprivileged user EXEC mode). Listing 4.2 demonstrates the `config terminal` command.

Listing 4.2 The config terminal Command

```
pixfirewall> enable
Password:
pixfirewall# config terminal
pixfirewall(config)# exit
pixfirewall# config terminal
pixfirewall(config)# disable
pixfirewall>
```

Monitor Mode

Monitor mode is symbolized by the `monitor>` prompt. This special mode enables you to perform maintenance features that are sometimes unavailable during normal operation. New binary images and password breaking procedure files can be downloaded in this mode. To enter monitor mode, reload your PIX. During the bootup phase, you will be prompted with this message: `Use BREAK or ESC to interrupt flash boot.`. Press either Break or ESC to enter monitor mode the 10-second timeout. Listing 4.3 is an example of the output displayed when entering monitor mode.

Listing 4.3 Monitor Mode

```
Cisco Secure PIX Firewall BIOS (4.2) #6: Mon Aug 27 15:09:54 PDT 2001
Platform PIX-501
Flash=E28F640J3 @ 0x3000000

Use BREAK or ESC to interrupt flash boot.
Use SPACE to begin flash boot immediately.
Flash boot interrupted.
0: i8255X @ PCI(bus:0 dev:17 irq:9 )
1: i8255X @ PCI(bus:0 dev:18 irq:10)

Using 1: i82557 @ PCI(bus:0 dev:18 irq:10), MAC: 000c.3085.5641
Use ? for help.
monitor>
```

Most CLI commands can be abbreviated, making your configuration tasks a little faster. For example, the command **enable** can be abbreviated to just **en** and the **config terminal** command can be just **con t**.

Be sure know your administrative access modes, which are as follows:

- *Unprivileged mode*—**pixfirewall>**
- *Privileged mode*—**pixfirewall#**
- *Configuration mode*—**pixfirewall(config)#**
- *Monitor mode*—**monitor>**

Knowing the General Commands

Several commands are covered in this section. These commands will help you monitor, display, and save your configurations, and they are all within the privileged or configuration mode. Therefore, use the `enable` and `config terminal` commands to enter the necessary access mode.

Here's a preview of the commands:

```
clear arp              reload
clear xlate            show arp
enable                 show conn
enable password        show history
hostname               show xlate
passwd                 telnet
ping
```

The `enable` Command

`enable` allows you to enter the privileged EXEC mode. Although this mode requires a password, the password is blank by default and simply pressing Enter when you see the password prompt lets you enter privileged EXEC mode.

The `enable password` command sets a privilege EXEC mode password. These passwords are case sensitive, so be careful. You can use the `show enable` command to display the encrypted version of the password stored in the configuration, like so:

```
pixfirewall(config)# enable password oregon
pixfirewall(config)# show enable
enable password W5TSthJO5zEtPi9F encrypted
pixfirewall(config)#
```

The passwd Command

The `passwd` command is used to set the password for Telnet access to the PIX. By default, this password is `cisco`; it must be in all lowercase because it's case sensitive. The following command sets the password to `cisco`:

```
pixfirewall(config)# passwd cisco
pixfirewall(config)#
```

The telnet Command

The `telnet` command specifies which hosts can connect to the PIX inside interface using Telnet. Telnet users can access the PIX on all interfaces except the outside interface. If users need access via the outside interface, an IPSec established connection is required before Telnet will connect. The Telnet syntax is as follows:

```
telnet <local_ip> [<mask>] [<if_name>]
```

Table 4.3 The telnet Command Options

Option	Function
local_ip	This is the IP address of the host you want to allow Telnet access.
mask	This is optional and can be used to define a whole subnet if necessary.
if_name	This is optional and is required only when you are using IPSec to connect to the outside interface.

The following example shows how to allow host 192.168.1.11 to Telnet into the PIX firewall on the inside interface:

```
pixfirewall(config)# telnet 192.168.1.11
pixfirewall(config)# show telnet
192.168.1.11 255.255.255.255 inside
pixfirewall(config)#
```

The hostname Command

The `hostname` command is used to change the command-line prompt as well as the fully qualified domain name used to generate RSA keys. The default

hostname is `pixfirewall`. The following command sets the hostname to `firewall2`:

```
pixfirewall(config)# hostname firewall2
firewall2(config)#
```

The show history Command

The `show history` command displays a list of previously entered commands. The following command displays the history:

```
pixtraincenter# show history
  show history
  show interface
  enable
```

The show conn Command

The `show conn` command displays connection table information about TCP traffic traveling through the PIX. In this example, host 192.168.1.11 with port 11969 is going to 165.193.123.44 port 80:

```
pixfirewall# show conn
1 in use, 5 most used
TCP out 165.193.123.44:80 in 192.168.1.11:1969 idle 0:00:03 Bytes 334
pixfirewall#
```

The show xlate Command

Use the `show xlate` command to view the current translation slots made in the translation table (recall that a PIX uses a connection table and a translation table to track the flow of traffic through its interfaces). *Translation slots* is the term used to describe the translation mapping from an internal address to a global external address. In this example, a local user using IP address 192.168.1.11 with port 1969 has been translated to a global outside interface address of 169.254.8.31 port 1237:

```
pixfirewall# show xlate
1 in use, 53 most used
PAT Global 169.254.8.31(1237) Local 192.168.1.11(1969)
PAT Global 169.254.8.31(2346) Local 192.168.1.12(5671)
pixfirewall#
```

The clear xlate Command

The `clear xlate` command clears the current translation slot entries. This should be done every time you add, modify, or delete something using the

following commands: `aaa-server`, `access-lists`, `alias`, `conduits`, `global`, `nat`, and `routes type`. This helps to reset the xlate table and make the previous command operate as expected. The following shows the command being executed from the privileged EXEC mode:

```
pixfirewall# clear xlate
```

The ping Command

The `ping` command enables you to test whether the PIX firewall can reach another IP address, and it results in a new mapping in the Address Resolution Protocol (ARP) table. The following example shows a `ping` command and a response:

```
pixfirewall# ping 192.168.1.11
    192.168.1.11 response received — 0ms
    192.168.1.11 response received — 0ms
    192.168.1.11 response received — 0ms
```

The **ping** command can be used to show that an IP address is reachable, but it doesn't test whether traffic can flow through the PIX firewall.

The show arp Command

The `show arp` command displays the ARP table, which maps an IP address to a physical MAC address. The following command displays the ARP cache:

```
pixfirewall# show arp
    inside 192.168.1.11 0002.a599.aa96
    inside 255.255.255.255 0002.a599.aa96
```

The clear arp Command

The `clear arp` command flushes all the entries in the ARP cache from RAM. The following command clears the ARP cache:

```
pixfirewall# show arp
    inside 192.168.1.11 0002.a599.aa96
    inside 255.255.255.255 0002.a599.aa96
pixfirewall# clear arp
pixfirewall# show arp
pixfirewall#
```

The reload Command

The `reload` command reboots the PIX firewall and loads the flash memory configuration into RAM. Please note that there is no such thing as a reboot command. The `reload` command displays the `reload` command and the resulting output from the command:

```
pixfirewall# reload
Proceed with reload? [confirm] y

Rebooting....

CISCO SYSTEMS PIX-501
Embedded BIOS Version 4.3.200 07/31/01 15:58:22.08
```

Viewing and Saving the Configuration

The ability to view and save the PIX firewall configuration is a vital part of setup and troubleshooting. The following section covers several of the most common commands. Here's a preview of them:

```
show configure        show version
show interface        write memory
show ip address       write standby
show startup          write terminal
```

The show startup Command

The `show startup` and `show configure` commands both display configurations saved in flash memory. These configurations are loaded into RAM during bootup. The following displays only the first eight lines of the output from the `show startup` command:

```
pixfirewall# show startup
: Saved
: Written by enable_15 at 04:55:12.917 UTC Wed Apr 2 2003
PIX Version 6.2(2)
nameif ethernet0 outside security0
nameif ethernet1 inside security100
enable password W5TSthJO5zEtPi9F encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname pixfirewall
```

The write terminal Command

You use the `write terminal` command to display the configuration currently running in RAM to the console. This configuration is also known as the *running config* and can be displayed using the `show running-config` command, as in other Cisco devices. This code displays the command's output:

```
pixfirewall# write terminal
: Saved
:
PIX Version 6.2(2)
nameif ethernet0 outside security0
nameif ethernet1 inside security100
enable password W5TSthJO5zEtPi9F encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname pixfirewall
```

The show interface Command

The `show interface` command displays information such as the IP address, line status, protocol status, and interface counter information. To display only one interface, add the hardware ID to the end of the command. Listing 4.4 displays the `show interface` output for interface Ethernet 1.

Listing 4.4 The show interface Output

```
pixfirewall(config)# show int e1
interface ethernet1 "inside" is up, line protocol is up
 Hardware is i82559 ethernet, address is 000c.3085.5641
 IP address 192.168.1.1, subnet mask 255.255.255.0
 MTU 1500 bytes, BW 10000 Kbit full duplex
    261 packets input, 32294 bytes, 0 no buffer
    Received 249 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    27 packets output, 3802 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 babbles, 0 late collisions, 0 deferred
    0 lost carrier, 0 no carrier
    input queue (curr/max blocks): hardware (128/128) software (0/1)
    output queue (curr/max blocks): hardware (0/2) software (0/1)
pixfirewall(config)#
```

The show IP address Command

You use the `show IP address` to display the address information assigned to each of the device's interfaces. The following command displays all the IP addresses assigned to the PIX firewall:

```
pixfirewall# show IP address
System IP Addresses:
    IP address outside 169.254.8.1 255.255.255.0
    IP address inside 192.168.1.1 255.255.255.0
```

```
Current IP Addresses:
    IP address outside 169.254.8.1 255.255.255.0
    IP address inside 192.168.1.1 255.255.255.0
pixfirewall#
```

The show version Command

The `show version` command enables you to view the firewall's software version, processor type, operating time since last reboot, flash memory type, interface boards, serial number, and activation keys. Listing 4.5 displays the output from the `show version` command.

Listing 4.5 The show version Command

```
pixfirewall# show version

Cisco PIX Firewall Version 6.2(2)
Cisco PIX Device Manager Version 2.1(1)

Compiled on Fri 07-Jun-02 17:49 by morlee

pixfirewall up 8 hours 31 mins

Hardware:  PIX-501, 16 MB RAM, CPU Am5x86 133 MHz
Flash E28F640J3 @ 0x3000000, 8MB
BIOS Flash E28F640J3 @ 0xfffd8000, 128KB

0: ethernet0: address is 000c.3085.5640, irq 9
1: ethernet1: address is 000c.3085.5641, irq 10
Licensed Features:
Failover:      Disabled
VPN-DES:      Enabled
VPN-3DES:      Disabled
Maximum Interfaces: 2
Cut-through Proxy: Enabled
Guards:       Enabled
URL-filtering:   Enabled
Inside Hosts:    10
Throughput:     Limited
IKE peers:     5

Serial Number: 807082785 (0x301b1b21)
Running Activation Key: 0x2d284af1 0xd032aa26 0x38b7db1f 0x70cfa8ee
Configuration last modified by enable_15 at 10:45:05.183 UTC Tue Apr 1 2003
pixfirewall#
```

The write memory Command

The `write memory` command saves the current running configuration to flash memory. When the system is reloaded, this configuration is loaded into RAM and executed as the running configuration. The following displays the command's syntax:

```
pixfirewall# write memory
Building configuration...
Cryptochecksum: 827c289b 6a6d8181 829b5b98 d3f1c82a
[OK]
pixfirewall#
```

Similarly, the `write standby` command saves the running configuration from the active PIX firewall to the standby PIX firewall when you are working with failover configurations. You can also think of this as writing from active RAM to standby RAM. Following is an example of the `write standby` command:

```
pixfirewall# write standby
```

The Six Basic Commands

The PIX firewall's basic setup is based on six primary commands. The commands shown in the following list provide the most basic configuration settings to allow traffic to flow through the firewall. This section covers each command in detail.

Here's a preview of the commands:

```
global          nameif

interface       nat

ip address      route
```

Naming Interfaces

Before we begin discussing these commands, a brief explanation is necessary to understand how interfaces are handled by the PIX. A name association needs to be designated for each hardware interface; it is this associated name rather than the hardware ID that is used in most of the configuration commands. For example, the interface `e1` is by default named `inside`. This name of `inside` is used throughout the PIX command structure as a pointer to the real hardware ID of interface `e1`.

Network Address Translation

Network address translation (NAT) is the process of translating multiple internal addresses to multiple global addresses. Every packet leaving the NAT translator uses the next available global address, and a translation table entry is made to record a link between the internal address and the outgoing global address. As packets flow back, the translated global address is reverted to the original

internal address. This is known as *dynamic mapping*, and the global addresses are only temporarily used. Table 4.4 displays the subnet of 192.168.1.0, which all share a global address pool of 169.254.8.31–169.254.8.35:

Table 4.4 NAT Mapping Table

NAT Internal Addresses	Global Address Pool
192.168.1.0 255.255.255.0	169.254.8.31–169.254.8.35

Table 4.5 shows a temporary mapping of the internal address of 192.168.1.11 to 169.254.8.31. If the internal host closes the session or loses the session, or the connection times out, 169.254.8.31 is released so another internal address can use it.

Table 4.5 Internal-to-Global Address Mapping

Internal Addresses	Globally Mapped Addresses
192.168.1.11	169.254.8.31 (temporary)
192.168.1.12	169.254.8.32 (temporary)
	169.254.8.33 (temporary)
	169.254.8.34 (temporary)
	169.254.8.35 (temporary)

For example, as Jack's computer talks to the Internet, his IP address of 192.168.1.11 is translated by the PIX using NAT to an address of 169.254.8.31 and subsequently passes the interface connected to the Internet. If another user, such as Timmy with an IP address of 192.168.1.12, is going through the PIX to the Internet, Timmy's IP address is translated to the next available global IP address, which is 169.254.8.32. This process continues to allocate the next available global IP address until none are left. At this point, a process of NAT *overloading*—also known as *PAT*—takes over. Figure 4.1 displays Jack's computer being translated to 169.254.8.31 as it travels through the PIX firewall.

Port Address Translation

Port address translation (PAT) is also called NAT overloading and is the process of translating multiple internal addresses to a single global address. Every packet leaving the PAT translator uses the same global address with a modified source port number. For example, as Jack's packet travels through the PIX, his IP address and port number are changed. An address of 192.168.1.11 port 1237 is modified to an address of 169.254.8.31 and the

next available port, such as port 5001. When Timmy requests information from the Internet, his address of 192.168.1.12 port 2403 is modified to the same 169.254.8.31 address but the port number is the next available port, such as 5002. When a request comes back from the Internet with 169.254.8.31 port 5001, this is referenced in the translation table to show that the packet should be changed back to 192.168.1.11 port 1237, and the packet is delivered to Jack.

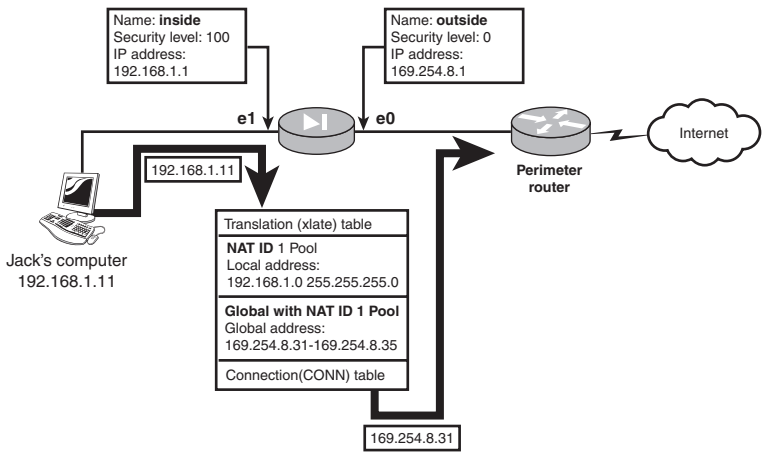

Figure 4.1 A NAT diagram.

Table 4.6 PAT Address Table

Nat Internal Address	Global Address Pool
192.168.1.0 255.255.255.0	169.254.8.31

Table 4.7 PAT IP Address and Port Mapping Table

Internal Addresses	Globally Mapped Addresses
192.168.1.11:**1237**	169.254.8.31:**5001** (temporary)
192.168.1.12:**1937**	169.254.8.31:**5002** (temporary)

Steps to Setting Up the PIX with the Six Basic Commands

1. `nameif`—Assign a name to a hardware ID interface and set the security level.

2. `interface`—Set the interface speed and enable the interface.
3. `ip address`—Assign an IP address to a named interface.
4. `nat`—Create a NAT ID that defines which local IP addresses will be translated on a specific named interface.
5. `global`—Create a global list of addresses to be used by the NAT ID in step 4.
6. `route`—Create any necessary static routes or default routes.

The nameif Command

The `nameif` command creates a name that is associated with a hardware interface and that is used throughout several other commands. Some examples of good names to use are `inside`, `outside`, and `DMZ`. The syntax of the `nameif` command is as follows:

```
nameif <hardware_id> <if_name> <security_lvl>
```

Table 4.8 nameif Options

Option	Function
hardware_id	The hardware ID is the name of the physical hardware, such as **e0** or **Ethernet0**.
if_name	This is the name you want to call the interface.
security_lvl	The security level setting is between 0 and 100. 0 is reserved for the outside interface, and 100 is used for the most secure inside interface.

The following example shows that hardware interface `Ethernet 1` is being set to `inside`:

```
pixfirewall(config)# nameif e1 inside security100
pixfirewall(config)#
```

The interface Command

The `interface` command sets the hardware speed and enables or disables an interface. Here's the syntax of the `interface` command:

```
interface <hardware_id> [<hw_speed> [<shutdown>]]
```

Table 4.9 interface Options	
Option	**Function**
hardware_id	The hardware ID is the name of the physical hardware, such as **e0** or **Ethernet0**.
hw_speed	This determines the connection speed used by this interface. The options are as follows: Auto—Autodetects the network speed 10BASE-T—10Mbp Ethernet half-duplex 10full—10Mbp Ethernet full-duplex 100BASE-TX—100Mbp Ethernet half-duplex 100full—100Mbp Ethernet full-duplex
shutdown	Defines whether the interface is administratively shut down.

The first command in the example enables the interface with 10BASE-T, and the second command disables the interface:

```
pixfirewall(config)# interface e1 10baseT
pixfirewall(config)# interface e1 10baseT shutdown
```

The ip address Command

The `ip address` command defines the layer 3 IP address on the interface and uses the name of the interface, as opposed to the hardware address. Its syntax is shown here:

```
ip address <if_name> <ip_address> [<mask>]
```

Table 4.10 ip adddress Options	
Option	**Function**
if_name	This is the name given to the hardware ID using the **nameif** command.
ip_address	This is the IP address you want to have on the address.
mask	This is the network mask.

In the following example, the inside interface (`e1`) is being set to an IP address of 192.168.1.1 and a subnet mask of 255.255.255.0:

```
pixfirewall(config)# ip address inside 192.168.1.1 255.255.255.0
```

The nat and global Commands

The `nat` and `global` commands work together to determine which addresses need translating and to what those addresses will be translated. NAT defines which addresses need to be translated. The ID field in the `nat` command corresponds to a `global` command that contains a pool of addresses used for translation. The `nat` command's syntax is shown here:

```
nat [(<if_name>)] <nat_id> <local_ip> [<mask> [dns] [outside]
              [<max_conns> [emb_limit> [<norandomseq>]]]]
```

Table 4.11 The nat Command's Options

Option	Function
if_name	This is the name of the internal interface to which this command is linked.
nat_id	This is the ID number that groups the **nat** command with the **global** command
local_ip	This defines which IP addresses are within this **nat_id** group.
mask	This defines the local_IP network mask.
dns	This specifies that the DNS replies that match xlate tables are translated.
outside	This specifies that the **nat** command applies to the outside interface.
max_cons	This defines the maximum number of TCP connections allowed.
emb_limit	This specifies the embryonic limit. The default is **0**, which is unlimited embryonic connections.
norandomseq	This states not to randomize the normal TCP packet sequence numbering.

The `global` command is used to allocate the address to which the internal address will be assigned. The syntax shown here details the `global` command:

```
global [(<ext_if_name>)] <nat_id> {<global_ip>[-<global_ip>]
[netmask <global_mask>]} ¦ interface
```

Table 4.12 The global Command's Options

Option	Function
ext_if_name	Defines the external interface on which these global addresses will be used.
nat_id	The ID number that links the **nat** command with the **global** address pool command.
global_ip	Assigns a single address or a pool of addresses to which the **nat** command will translate its internal address.
interface	If you use this option, the interface is to set up to use PAT or NAT overloading on the same IP address assigned to the interface.

In Listing 4.6, the address 192.168.1.20 on the inside interface is translated to an IP address of 169.254.8.5 on the outside interface. These two commands are linked by the `nat_id` of `12`.

Listing 4.6 The nat and global Commands for a Single Host

```
pixfirewall(config)# nat (inside) 12 192.168.1.20 255.255.255.255
pixfirewall(config)# global (outside) 12 169.254.8.5 255.255.255.0
```

In Listing 4.7, the network of 192.168.1.0 255.255.255.0 on the inside interface is translated to a global pool of addresses 169.254.8.10–169.254.8.20 on the outside interface. These two commands are linked by the `nat_id` of `5`.

Listing 4.7 The nat and global Commands for a Subnet

```
pixfirewall(config)# nat (inside) 5 192.168.1.0 255.255.255.0
pixfirewall(config)# global (outside) 5 169.254.8.10-169.254.8.20
```

In Listing 4.8, all the addresses on the inside interface are translated to the global address that is defined as the outside interface IP address. This many-to-one solution uses PAT.

Listing 4.8 nat and global Commands

```
pixfirewall(config)# nat (inside) 1 0.0.0.0 0.0.0.0
pixfirewall(config)# global (outside) 1 interface
```

In Listing 4.9, three networks on different interfaces are all part of the `nat_id 3` group. The `global` command linked to `nat_id 3` defines an address range of 168.254.8.5–168.254.8.10 to be used.

Listing 4.9 The nat and global Commands for Multiple Interfaces

```
pixfirewall(config)# nat (inside) 3 192.168.1.0 255.255.255.0
pixfirewall(config)# nat (dmz) 3 192.168.2.0 255.255.255.0
pixfirewall(config)# nat (dmz2) 3 192.168.3.0 255.255.255.0
pixfirewall(config)# global (outside) 3
                168.254.8.5-168.254.8.10 netmask 255.255.255.0
```

You can use the `show nat` and `show global` commands to display the list of NAT and global entries made, and you can use the `no nat` and `no global` commands to disable the entries made.

The route Command

The `route` command is used to add a static or default route to an interface. This syntax shows the command:

```
route <if_name> <foreign_ip> <mask> <gateway> [<metric>]
```

Table 4.13 route Command Options	
Option	**Function**
if_name	This is the internal or external interface name the traffic will use to exit from the PIX.
foreign_ip	The foreign IP is the destination network address. To define a default route, you can use 0.0.0.0 as the address and 0.0.0.0 as the mask.
gateway	This is the gateway, which is also known as the *next hop router*.
metric	This is the metric value used to define the number of hops away the destination network is.

In this example a default route has been created that will forward traffic to a router at 169.254.8.100 with a metric of 1:

```
pixfirewall(config)# route outside 0.0.0.0 0.0.0.0 169.254.8.100 1
```

Using the Six Commands

For the exam, be sure you cover this section and know these commands to configure a PIX firewall. Listing 4.10 shows the commands needed to configure your PIX firewall according to Figure 4.2.

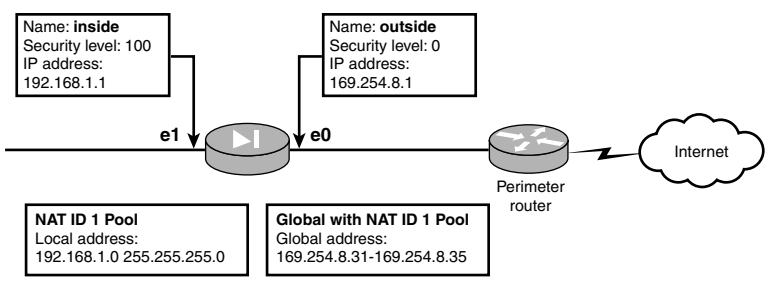

Figure 4.2 NAT diagram.

Listing 4.10 Sample Code

```
pixfirewall# config t
pixfirewall(config)# nameif e0 outside security0
pixfirewall(config)# nameif e1 inside security100

pixfirewall(config)# interface e0 10baseT
pixfirewall(config)# interface e1 10full

pixfirewall(config)# ip address outside 169.254.8.1 255.255.255.0
pixfirewall(config)# ip address inside 192.168.1.1 255.255.255.0
```

Listing 4.10 Sample Code *(continued)*

```
pixfirewall(config)# nat (inside) 1 192.168.1.0 255.255.255.0
pixfirewall(config)# global (outside) 1 169.254.8.31-169.254.8.35

pixfirewall(config)# route outside 0.0.0.0 0.0.0.0 169.254.8.100 1
```

Working with the Trivial File Transfer Protocol

PIX firewalls can save, restore, and install new images from Trivial File Transfer Protocol (TFTP) servers. TFTP servers enable clients such as the PIX firewall to save and read files, similar to the way in which normal FTP functionality allows clients to download and upload files on the Internet. Several TFTP programs are available on the market, and Cisco provides a simple TFTP server for free with the PIX firewall. Alternatively, the TFTP server can be downloaded from the Cisco Web site.

Images are typically operating system upgrades or PDM images needing to be uploaded to the PIX.

A general TFTP command can be used to help make the other commands a little shorter.

The command is **tftp-server [if_name] ip_address path** and it enables you to enter the default TFTP server parameters used on other commands.

Upgrading OS Images

There are two methods of upgrading a new OS image to the PIX firewall. The first method uses the `copy` command or booting to `monitor>` mode.

The `copy` Command

This command is available on IOS versions 5.1 and 5.3 and above, and it is used to download images from a TFTP server to the firewall. The TFTP option of the command is for the location and path of the image, whereas the flash option determines whether it's an image or PDM software. The `copy` command's syntax is as follows:

```
copy tftp[:[[//location] [/pathname]]] flash[:[image ¦ pdm]]
```

Table 4.14 copy Command Options

Option	Function
tftp	This option allows for the location and path of the image you want to download.
flash	This option enables you to specify which type of image you are downloading: an image for a new IOS or a PDM for Cisco's graphical user interface.

Most Cisco IOSes use the **copy** command in one form or another. One of the most common problems when remembering the **copy** command syntax is the order of the parameters. An easy way to remember the order is the phrase, *copy from to*, or just CFT, which is alphabetical. This means copy from some location to some destination.

The following is the step-by-step process you would use to copy an image from a TFTP server to a PIX firewall. Figure 4.3 displays the networking layout, and Listing 4.11 shows the necessary commands. Follow these steps:

1. Start the TFTP program on your server; this example uses 192.168.1.11 as the server.
2. Enter the `copy tftp flash` command.
3. At the prompt, enter the TFTP server IP address—for example, enter `192.168.1.11`.
4. At the prompt, enter the source filename—for example, enter `pix622.bin`.
5. Enter `yes` to continue. This starts the download of the image to the PIX firewall.
6. Reload the PIX and enjoy your new OS!

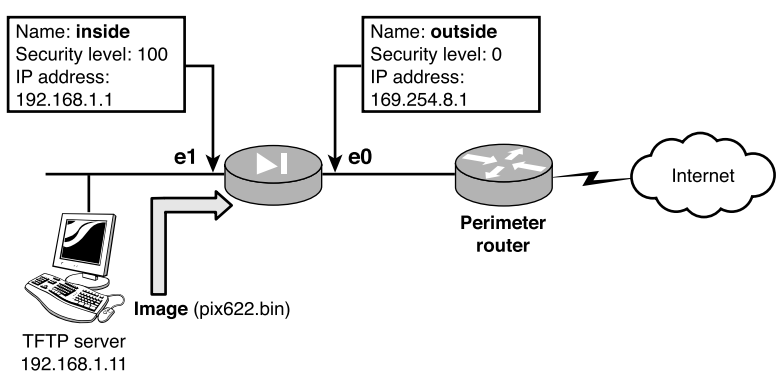

Figure 4.3 TFTP diagram.

Listing 4.11 Copying from a TFTP Server

```
Pixfirewall# copy tftp flash
Address or name of remote host []? 192.168.1.11
Source file name []? pix622.bin
copying tftp://192.168.1.11/pix622.bin to flash:image
[yes¦no¦again]? yes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Received 1658880 bytes
Erasing current image
Writing 1540152 bytes of image
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Image installed
pixfirewall#
```

Using Monitor Mode to Upgrade Images

In the past, using monitor mode was the only way you could upgrade your OS images. However, this process has now been replaced by the `copy` command. Follow these step-by-step instructions on how to upload an image in monitor mode:

1. Start the TFTP program on your server; this example uses 192.168.1.11 as the server
2. Restart your PIX by cycling power or using the `reload` command.
3. Press Break or ESC to interrupt the flash boot and enter into monitor mode.
4. Enter the interface you want to use—for example, enter **`monitor> interface 11`**.
5. Enter the interface IP address—for example, enter **`monitor> address 192.168.1.1`**.
6. Enter the default gateway, if required—for example, enter **`monitor> gateway IP address`**.
7. Test communication with the TFTP server using the `ping` command—for example, enter **`monitor> ping 192.168.1.11`**.
8. Enter the TFTP server IP address—for example, enter **`monitor> server 192.168.1.11`**.
9. Enter the image's filename—for example, enter **`monitor> file pix622.bin`**.
10. Begin the TFTP process by entering the keyword **`tftp`**.
11. When the upload is done, enter **`y`** to copy the image to flash.
12. Reload the PIX.

Listing 4.12 displays the monitor mode and TFTP steps needed to upload an image to your PIX.

Listing 4.12 Monitor Mode's tftp Command

```
monitor> interface 1
monitor> address 192.168.1.1
address 192.168.1.1
monitor> ping 192.168.1.11
monitor> server 192.168.1.11
server 192.168.1.11
monitor> file pix622.bin
file pix622.bin
monitor> tftp
.................................................................
Received 1658880 bytes

Cisco Secure PIX Firewall admin loader (3.0) #0:
Fri Jun 7 17:35:02 PDT 2002
Flash=E28F640J3 @ 0x3000000
BIOS Flash=E28F640J3 @ 0xD8000
Flash version 6.2.2, Install version 6.2.2
Do you wish to copy the install image into flash? [n] y

Installing to flash

Serial Number: 807082785 (0x301b1b21)
Activation Key: 2d284af1 d032aa26 38b7db1f 70cfa8ee

Do you want to enter a new activation key? [n]n
Writing 1540152 bytes image into flash...
```

Configuration Files

Saving, erasing, and restoring your configurations are all basic functions you will need to master. All the following examples assume you already have a TFTP server running.

The write net Command

Making a backup of your configuration is always a good idea because it makes recovery easy. You can save your configuration to a TFTP server using the `write net` command, like so:

```
write net [[server_ip]:[filename]]
```

Table 4.15 The write net Command Options	
Option	**Function**
server_ip	This is the IP address of the TFTP server.
filename	This is the path and name you want to save the configuration as.

Here is an example of the `write net` command:

```
pixfirewall# write net 192.168.1.11:backup1
Building configuration...
TFTP write 'backup1' at 192.168.1.11 on interface 1
[OK]
```

The write erase Command

The `write erase` command erases your configuration from flash, giving you an empty configuration on the next reload. Here's a command example:

```
pixfirewall(config)# write erase
Erase PIX configuration in flash memory? [confirm]
```

The configure net Command

Finally, the `configure net` command enables you to merge your configuration back into the PIX from a TFTP server. Make a note that you must configure at least one interface with an IP address and enable it before you can reload your configuration. Here's its syntax:

```
configure net [<location>]:[<pathname>]
```

Table 4.16 configure net Command Options	
Option	**Function**
location	The location option is the IP address of the TFTP server.
pathname	This is the path and filename of the configuration.

The `write net` command's syntax is as follows:

```
pixfirewall# write net 192.168.1.11:backup1
Building configuration...
TFTP write 'backup1' at 192.168.1.11 on interface 1
[OK]
```

Password Recovery

If you forget your enable password, the PIX firewall requires you to upload a file to the flash. This special file nullifies the current password without erasing your configuration. The process is virtually identical to loading a new image using the `monitor>` prompt and a TFTP server. You can download the password file for your specific version of OS image at `www.cisco.com/warp/public/110/34.shtml`.

The password file for 6.2 release, for instance, is `np62.bin`. This utility resets the enable and Telnet passwords to their default settings, which is `cisco` for both of them.

Listing 4.13 shows the steps for uploading a password recovery file.

Listing 4.13 Password Recovery Example

```
monitor> interface 1
monitor> address 192.168.1.1
address 192.168.1.1
monitor> server 192.168.1.11
server 192.168.1.11
monitor> file np62.bin
file np62.bin
monitor> tftp
tftp np62.bin@192.168.1.11.........................
Received 73728 bytes

Cisco Secure PIX Firewall password tool (3.0) #0:
Wed Mar 27 11:02:16 PST 2002
Flash=E28F640J3 @ 0x3000000
BIOS Flash=E28F640J3 @ 0xD8000

Do you wish to erase the passwords? [yn] y
The following lines will be removed from the configuration:
    enable password ZFatiF0MarNtVoTD encrypted
    passwd 2KFQnbNIdI.2KYOU encrypted

Do you want to remove the commands listed above
from the configuration? [yn] y
```

Password recovery on older PIX firewalls such as the PIX 510 and 520 is done using a floppy disk: A password lockout utility is loaded from a floppy, and the PIX firewall is rebooted.