Configuring the PIX Firewall

You can configure the PIX Firewall by entering commands similar to those of Cisco IOS technology.

When shipped from Cisco, each PIX Firewall comes with a basic configuration that lets the unit boot up, but does not let network traffic pass through until you configure it to do so.

This chapter describes how to start a configuration and build on it. Table 2-1 lists the sections in this chapter. The material is presented as a series of steps that you can follow completely if you are creating a new configuration, or as needed with an existing configuration.

Table 2-1 Chapter Topics

Before Configuring PIX Firewall	Initial Configuration	Continuing
Step 1 - Get a Console Terminal	Step 5 - Identify Each Interface	Step 12 - Add Telnet Console Access
Step 2 - Get the Most Current Software	Step 6 - Let Users Start Connections	Step 13 - Add Server Access
Step 3 - Configure Network Routing	Step 7 - Create a Default Route	Step 14 - Add Static Routes
Step 4 - Start Configuring PIX Firewall	Step 8 - Permit Ping Access	Step 15 - Enable Syslog
	Step 9 - Store the Image in Flash Memory and Reboot	Step 16 - Create Access Lists
	Step 10 - Check the Configuration	Step 17 - Add AAA User Authentication
	Step 11 - Test Network Connectivity	Step 18 - Recheck the Configuration

Information in Steps 2 and 4 overlap with the initial configuration information in the *Installation Guide for the Cisco Secure PIX Firewall Version 5.0*, but are shown here to provide continuity.

Acronyms in this chapter are defined in Appendix B, "Acronyms and Abbreviations." All commands shown in this chapter are explained fully in Chapter 6, "Command Reference."

Upgrading from a Previous Version

Before upgrading from a previous version, save your configuration and write down your activation key. Information for upgrading the failover feature is described in the "Failover" section in Chapter 3, "Advanced Configurations."

Step 1 - Get a Console Terminal

If the computer you are connecting to runs either Windows 95 or Windows NT, the Windows HyperTerminal accessory provides easy-to-use software for communicating with the firewall. If you are using UNIX, refer to your system documentation for a terminal program.

HyperTerminal also lets you cut and paste configuration information from your computer to the firewall console.

To configure HyperTerminal:

Step 1 Connect the serial port of your PC to the console port of the PIX Firewall with the serial cable supplied in the PIX Firewall accessory kit.

Step 2 Locate HyperTerminal by opening the Windows 95 or Windows NT **Start** menu and clicking **Programs>Accessories>HyperTerminal**.

Step 3 Double-click the Hypertrm accessory. The New Connection window opens with the smaller Connection Description dialog box in the center.

Step 4 Enter the name of the connection. You can use any name such as PIX Console. Click **OK** when you are ready to continue.

Step 5 At the Phone Number dialog box, ignore all the fields except "Connect using." In this field, click the arrow at the right to view the choices. Click "Direct to Com 1," unless you are using another serial port. Click **OK** to continue.

Step 6 At the COM1 Properties dialog box, set the following fields:

- Bits per second to 9600.
- Data bits to 8.
- Parity to None.
- Stop bits to 1.
- Flow control to Hardware.

Step 7 Click **OK** to continue.

Step 8 The HyperTerminal window is now ready to receive information from the PIX Firewall console. If the serial cable is connected to the firewall, power on the firewall and you should be able to view the console startup display.

If nothing happens, wait 60 seconds first. The firewall does not send information for about 30 seconds. If messages do not appear after 60 seconds, press the **Enter** key. If still nothing appears, ensure that the serial cable is attached to COM1 and not to COM2 if your computer is so equipped. If garbage characters appear, ensure that the bits per second setting is 9600.

Step 9 On the **File** menu, click **Save** to save your settings.

Step 10 On the **File** menu, click **Exit** to exit HyperTerminal. HyperTerminal prompts you to be sure you want to disconnect. Click **Yes**.

HyperTerminal saves a log of your console session that you can access the next time you use it.

To restart HyperTerminal, double-click the connection name you chose in the HyperTerminal folder. When HyperTerminal starts, drag the scroll bar up to view the previous session.

Step 2 - Get the Most Current Software

This section includes the following topics:

- Latest Software
- Download over the Web
- Download with FTP
- Creating a Bootable Diskette from Windows
- Creating a Bootable Diskette from UNIX

Latest Software

If desired, you can obtain the most current version of the PIX Firewall software by downloading it from Cisco's online web or FTP site. If you are using FTP, refer to the section "Download with FTP." If you are using the Web, refer to the section "Download over the Web." The sections that follow describe how to download the software and prepare a PIX Firewall bootable diskette. When the diskette is ready, you can insert it in the PIX Firewall's diskette drive and restart the firewall. This will give you access to the most current software on your PIX Firewall.

The files you can download follow:

- **.bin**—For UNIX, or for Windows and Windows NT if you already have the rawrite.exe program. Refer to "Creating a Bootable Diskette from UNIX" for installation information. If you have a PIX 515, you can put the .bin image on a TFTP server and download it to the PIX 515—refer to Chapter 7, "PIX 515 Configuration" for information on how to download the image to the PIX 515.
- **.exe**—For Windows and Windows NT. Except for the rawrite.exe program for creating bootable diskettes, the rest of the .exe files are self-extracting archives. Refer to the *Installation Guide for the Cisco Secure PIX Firewall Version 5.0* for information on installing the PFSS, and PFM. Refer to "Creating a Bootable Diskette from Windows" for installation information about the pix50*n*.exe and rawrite.exe files.

  These files are:

  — **pix50*n*.exe**—Contains the PIX Firewall image, instructions, and the rawrite.exe program.

  — **pfss422.exe**—Contains the PIX Firewall Syslog Server (PFSS), which provides a Windows NT Server that receives syslog messages from the PIX Firewall and stores them in daily log files. The PIX Firewall sends messages to the PFSS via TCP or UDP and can receive syslog messages from up to 10 PIX Firewall units. The version 4.4(2) PFSS works with versions 4.4, 5.0, and later.

  — **pfm432c.exe**—Contains the PIX Firewall Manager (PFM) and its accompanying files. As an alternative to the PFSS, the PFM GUI (graphical user interface) lets you manage up to 10 PIX Firewall units. The PFM also contains a syslog server that must not be used with the PFSS. Version 4.3(2)c or later of the PFM accepts PIX Firewall versions 4.3, 4.4, 5.0, and later. The PFM has not been upgraded with version 5.0 changes. Refer to the *Release Notes for the PIX Firewall Manager Version 4.3(2)c* for more information on how to install and use this feature.

  — **psw501.exe**—Contains the PIX Firewall Setup Wizard, which simplifies the PIX Firewall installation. The Setup Wizard works with PIX Firewall versions 4.3, 4.4, 5.0 and later. Refer to the *Installation Guide for the Cisco Secure PIX Firewall Version 5.0* for how to install the Setup Wizard.

  — **rawrite.exe**—A program you use to create a bootable diskette for the PIX Firewall.

Download over the Web

To download PIX Firewall software from the CCO web site:

Step 1 Use a network browser, such as Netscape Navigator to access http://www.cisco.com.

Step 2 If you are a registered CCO user, click **LOGIN** in the upper area of the page. If you have not registered, click **REGISTER** and follow the steps to register.

Step 3 After you click **LOGIN**, a dialog box appears requesting your Username and Password. Enter these and click **OK**.

Step 4 When you are ready to continue, choose **Software Center** under the **Service & Support** heading.

Step 5 On the Service & Support page, click **Internet Products** from the center column.

Step 6 On the Internet Products page, scroll down to the **Other Internet Software** bullet item. Then scroll down further and click **PIX Firewall Software**.

Step 7 On the PIX Firewall Software page, click **Download PIX Firewall Software**.

Step 8 On the software download page, choose the software you need depending on the file suffix: .exe or .bin as described in the last section.

Step 9 The Software Download page appears and provides these choices:

(a) Choice 1—To copy the file directly to your hard drive, choose a regional site closest to your location. A dialog box appears requesting that you enter your CCO password again. Enter it and click **OK**. The Save As dialog box appears and lets you specify the directory and output filename of the file on your hard drive. You can store the executable file anywhere. When executed, it will extract three files into the same directory in which it is run.

Choose the directory and filename and click **Save**. A dialog box appears to show you the progress of the transfer.

(b) Choice 2—If you want to receive the file by email, enter the destination email address and the file will be encoded with the UNIX **uuencode** command before being sent to the address you specify.

(c) Choice 3—Cisco Support engineers can give you access to the file via FTP. You can also use FTP to access this site directly.

Download with FTP

Before using FTP, you need to have previously registered with Cisco, which you can do via the Web or by calling Cisco.

Set your FTP client for passive mode. If you are not running in passive mode, you can log in and view the Cisco presentation messages, but entering commands will cause your client to appear to suspend execution.

The Windows 95 and Windows NT command line FTP programs do not support passive mode.

To get the most current software with FTP:

Step 1 Start your FTP client and connect to **cco.cisco.com**. Use your CCO username and password.

Step 2 You can view the files in the main directory by entering the **ls** command.

Step 3 Enter the **cd cisco** command to move to the cisco directory. Then enter **cd internet** and **cd pix** to access the PIX Firewall software directory. Use the **ls** command to view the directory contents.

Step 4 Use the **get** command to copy the proper file to your workstation as described at the start of the current section. If you want documentation, use the **cd documentation** command from the pix directory and copy the files you need to your workstation. Files with the .pdf suffix can be viewed with Adobe Acrobat Reader, which you can download from:

http://www.adobe.com/prodindex/acrobat/readstep.html

Step 5 When you are done, use **quit** to exit.

Creating a Bootable Diskette from Windows

Step 1 Using Windows Explorer or My Computer, open a window to the directory containing the archive and double-click the filename of the .exe file. It will automatically execute and provide these files:

- pix5*nn*.bin—The PIX Firewall binary file, where 5 is the version number and *nn* is the release number.
- rawrite.exe—The conversion utility that creates a PIX Firewall bootable diskette.
- readme.txt—Contains instructions about how to create the bootable diskette.

A sample archive extraction follows:

```
...extraction utility messages...
Searching EXE: C:/PIX/PIX5nn.EXE
  Inflating: README.TXT
  Inflating: PIX5nn.BIN
  Inflating:  RAWRITE.EXE
```

Step 2 Locate an IBM formatted diskette that does not contain useful files. Do not use the PIX Firewall boot diskette that came with your original PIX Firewall purchase—you will need this diskette for system recovery should you need to downgrade versions.

The rawrite program erases all the files on the diskette. If you format the diskette from Windows, choose the long version, not the quick format. The quick format does not adequately prepare the diskette for rawrite. The best way to format the diskette is from the MS-DOS command prompt.

Step 3 Enter **rawrite** at the MS-DOS command prompt and you are prompted for the name of the .bin binary file, the output device (**a:** or **b:** for a 3.5-inch diskette), and to insert a formatted diskette.

The utility then creates a PIX Firewall boot diskette.

A sample **rawrite** session follows:

```
C:\pix>rawrite
RaWrite 1.2 - Write disk file to raw floppy diskette

Enter source file name: pix5nn.bin
Enter destination drive: a:
Please insert a formatted diskette into drive A: and press -ENTER- :
Number of sectors per track for this disk is 18
Writing image to drive A:. Press ^C to abort.
Track: 78 Head: 1 Sector: 16
Done.
C:\pix>
```

Note Ensure that the binary filename is in the "8.3" character format (8 characters before the dot; 3 characters after the dot). Due to the size of the version 5.0 image, creating a diskette may take several minutes to complete.

Step 4 Remove the diskette from the drive, place it in the PIX Firewall diskette drive and power cycle the unit. Alternately, if your unit has a Reset switch, use it, or you can enter the **reload** command from the PIX Firewall console. The PIX Firewall then boots from the new diskette.

To continue the configuration, proceed to "Step 3 - Configure Network Routing."

Creating a Bootable Diskette from UNIX

Step 1 Download the .bin binary file to your local directory.

Step 2 Insert a diskette in your workstation's diskette drive.

Step 3 Enter the following command to copy the binary file to the diskette:

```
# dd bs=18b if=./pix5nn.bin of=/dev/rfd0
```

This command copies the binary file to the output device file with a block size of 18 blocks.

Note The diskette may have a name other than rfd0 on some UNIX systems.

Step 4 Eject the diskette, insert it in the PIX Firewall diskette drive, and power cycle the unit. Alternately, if available, use your unit's Reset switch, or enter the **reload** command from the PIX Firewall console. The PIX Firewall then boots from the new diskette.

When done, continue your configuration with "Step 3 - Configure Network Routing."

Step 3 - Configure Network Routing

Read this section before configuring the PIX Firewall to help you make decisions for configuring network routing.

This section includes the following topics:

- Preparing Routers to Work with the PIX Firewall
- Setting a Default Route for Each Host

Routing directs the flow of packets through a network. A default route specifies to which router packets are sent when the address is not known.

A host sends a message to another user. If the computer itself does not contain a login account for the user, the computer sends the message to its default gateway router. A router stores the paths through the network known as routes. If a router does not have the route to the user in its storage, it passes the message to its default router which knows routes from the larger network. The message is checked against the routes in this router. If it is not found, it is sent to another router with a still larger view of the network. This process repeats with the message sent from one router to another until the message is sent to the correct destination.

Preparing Routers to Work with the PIX Firewall

Once you have configured the PIX Firewall, you need to configure the other devices that will interact with the PIX Firewall. The most important element that works with the PIX Firewall are the routers, or switches, if they have routing capability. The instructions that follow assume that the routers are from Cisco.

To prepare the routers to work with the PIX Firewall:

Step 1 Connect a computer to the console port of the router that connects to the outside interface of the PIX Firewall. If you are using a Windows PC, you can use the HyperTerminal program with the router as well. You will need to know the username and password for the router.

Step 2 Access configuration mode by entering the **configure terminal** command.

Step 3 Clear the ARP cache. Use the **clear arp** command. Then enter **Cntrl-Z** to exit configuration mode.

Step 4 Connect to the router on the inside of the PIX Firewall and access configuration mode.

Step 5 Set the default route to the inside interface of the PIX Firewall with the following command:

```
ip route pix_inside_interface_ip_address
```

Step 6 Enter the **show ip route** command and make sure that the PIX Firewall interface is listed as the "gateway of last resort."

Step 7 Clear the ARP cache with the **clear arp** command. Then enter **Cntrl-Z** to exit configuration mode.

Step 8 If you changed the default route, use the **write memory** command to store the configuration in Flash memory. The **clear arp** command will make the new default gateway usable by the router.

Step 9 Connect to the routers on each perimeter interface and repeat the commands in Steps 5 through 8 for each router.

Step 10 If you have routers on networks subordinate to the routers that connect to the PIX Firewall's interfaces, configure them so that their default routes point to the router connected to the PIX Firewall and then clear their ARP caches as well.

Because the PIX Firewall is not a router, you need to specifically tell it where to route packets. The PIX Firewall lets you specify one default route to the outside interface, with one exception: if your PIX Firewall has only two interface cards installed, you can specify two default routes, one for the outside and one for the inside.

Note For a PIX Firewall with 3 or more interfaces, only the outside default route is allowed.

In many networks, the interface connecting to the PIX Firewall connects to a router. Many times, a number of networks connect to the router. To ensure that the PIX Firewall can see these routes, you need to add static **route** command statements for each network.

Both default and static routes are set on the PIX Firewall with the **route** command.

Setting a Default Route for Each Host

Each host on the same subnet as the inside or perimeter interfaces must have its default route pointing to the PIX Firewall.

This section includes the following topics:

- Setting a Solaris or SunOS Default Route
- Setting a LINUX Default Route
- Setting a Windows 95 and Windows 98 Default Route
- Setting a Windows NT Default Route
- Setting a MacOS Default Route

Setting a Solaris or SunOS Default Route

If the host is a Solaris or SunOS workstation, you can determine the default route with this command:

```
netstat -nr
```

With root permissions, edit the /etc/defaultrouter file to point the default route at the PIX Firewall and then reboot the workstation so that the information is usable.

Setting a LINUX Default Route

On LINUX systems, use the **netstat -r** command to view the routing table including the default route.

With root permissions, use the following command to set the default route:

```
route add default gw IP_address_of_next_host
```

Replace *IP_address_of_next_host* with the IP address of the next host.

Setting a Windows 95 and Windows 98 Default Route

If the host is a Windows workstation, you can view the default route by clicking **Start>Run** and entering this command:

```
winipcfg
```

To change the default route, click **Start>Settings>Control Panel** and double-click the **Network** item.

Select the TCP/IP entry from the list of installed network components and click **Properties**. The default route is on the Gateway tab.

Setting a Windows NT Default Route

You can view the default route from the Command Prompt by entering the **ipconfig** command. You can access the Command Prompt by clicking **Start>Programs>Command Prompt**.

To change the default gateway in Windows NT:

Step 1 Click the **Protocols** tab.

Step 2 In the Network Protocols window, click **TCP/IP Protocol**, and click **Properties**.

Step 3 In the Microsoft TCP/IP Properties window, click the **IP Address** tab.

Step 4 Click **Advanced** The default gateway IP address appears in the Gateways window. If the gateway is not the address of the PIX Firewall interface to which the server is connected, select the gateway address and click **Remove**.

Step 5 Click **Add** and enter the IP address for the PIX Firewall interface.

Step 6 After you exit from the menus, Windows will prompt you to restart your computer. Click **Yes**.

Setting a MacOS Default Route

You can view the default route from the MacOS 7.5 and later from the **Apple menu>Control Panels>TCP/IP** window. You can also set the default route from this window.

Step 4 - Start Configuring PIX Firewall

Before continuing, view "Command Line Guidelines" in Chapter 1, "Introduction," for information on how to specify ports and protocols, terminology, and other useful PIX Firewall facts.

When you start your PIX Firewall for the first time or load a new PIX Firewall boot disk, the configuration comes with many of the commands you need to get started. The configuration you first receive is known as the default configuration and is described in more detail in Chapter 1, "Introduction."

You can use the **write terminal** command to view your configuration at any time. Use the **write memory** command frequently to save your configuration to Flash memory.

Before you configure the PIX Firewall, sketch out a network diagram with IP addresses that you will assign to the PIX Firewall and those of routers on each interface. If you have more than two interfaces in the PIX Firewall, note the security level for each interface. Security levels are set with the **nameif** command described in "Step 5 - Identify Each Interface."

Locate the following IP addresses:

- An IP address for each interface that will connect to a network segment. Each address must be unique so that it is not used in the pool of global addresses or with any other command statement in the configuration.
- A pool of global addresses for each interface that each translated connection uses as it passes through the firewall. Use a global pool to let users start connections from a higher security level interface to access a lower security level interface.
- The IP address of the outside default router.

Go to the PIX Firewall Configuration Mode

To initially configure the PIX Firewall:

Step 1 Start your terminal emulation program.

Step 2 Power on the PIX Firewall. On newer models, the switch is at the back, on older models, the front.

Step 3 If you are configuring a PIX 515 and your site downloads configuration images from a central source with TFTP, look for the following prompt in the startup messages:

```
Use BREAK or ESC to interrupt flash boot.
```

PIX Firewall holds this prompt for 10 seconds. To download an image, press the Escape key to start boot mode. If you are not downloading an image, ignore the prompt or press the Space bar to start immediately and PIX Firewall starts normally. Refer now to Chapter 7, "PIX 515 Configuration" for information about how to download the configuration image.

Step 4 After the startup messages appear, you are prompted with the following unprivileged mode prompt:

```
pixfirewall>
```

Enter **enable** and press the **Enter** key.

Step 5 The following prompt appears:

```
Password:
```

Press the **Enter** key.

Step 6 You are now in privileged mode. The following prompt appears:

```
pixfirewall#>
```

Enter the **configure terminal** command and press **Enter**. You are now in configuration mode.

Step 5 - Identify Each Interface

On new installations, PIX Firewall provides names for each interface, which you can view with the **show nameif** command. If you want to provide alternative names, use the **nameif** command to do so.

This section includes the following topics:

- Two-Interface PIX Firewall
- Three or More Interfaces in the PIX Firewall

For new installations, PIX Firewall requires that you enable the use of each interface you intend to use with the **interface** command.

You need to specify a unique IP address for each interface you want to use with the **ip address** command.

Before deciding how to identify each interface, you should be sure you have the best network connected to meet your needs. Refer to the section "Deciding How to Use Multiple Interfaces" in Chapter 1, "Introduction."

Refer to the *Installation Guide for the Cisco Secure PIX Firewall Version 5.0* for a description of the various configurations that can occur depending on in which slot a 4-port card resides. Using a PIX 515 or PIX 520 changes how the unit determines how each network connects to the PIX Firewall.

The **nameif** Command

The PIX Firewall default configuration supplies **nameif** commands for the inside and outside interfaces. Use the **show nameif** command to view these commands. They will appear as follows:

```
nameif ethernet0 outside security0
nameif ethernet1 inside security100
```

The **nameif** commands you need to enter, if any, are determined by how many network interface cards are in your PIX Firewall. The sections that follow describe how to configure this command.

An example **nameif** command is:

```
nameif ethernet2 perimeter security50
```

If you make a mistake or want to replace a command you entered, enter the new version of the command, instead of first removing the old version, as is required for other PIX Firewall commands. For example, if you accidentally enter:

```
nameif ethernot2 permetter security50
```

Reenter the command as:

```
nameif ethernet2 perimeter security50
```

Two-Interface PIX Firewall

If you have only two interfaces, you do not need to enter any further information for the **nameif** command and can now proceed to next command for your configuration.

Three or More Interfaces in the PIX Firewall

PIX Firewall provides **nameif** commands for all interfaces. The inside interface is named "inside" and the outside interface is named "outside." Any perimeter interfaces are named "intf*n*," such as "intf2" for the first perimeter interface, "intf3" for the second perimeter interface up to a maximum of "intf5" for the fourth perimeter interface (PIX Firewall supports up to 6 interfaces). The numbers correspond to the interface card's position in the PIX Firewall, such that for Ethernet interfaces, ethernet0 is the outside interface, ethernet1 is the inside interface, ethernet2 is the first perimeter interface, and so on up to ethernet5 as the fourth perimeter interface. You can use the default names or give each interface a more meaningful name.

The format for the command is:

nameif *hardware_id interface security_level*

where:

- *hardware_id*—The hardware name for the network interface card. If you have all Ethernet interfaces in the PIX Firewall, use **ethernet2** and **ethernet3** for the **nameif** commands you supply.

  If you have both Ethernet and Token Ring cards, the third and fourth interfaces' *hardware_id* names differ depending on the interface type. For example, if you have an Ethernet interface on the outside, a Token Ring on the inside, and an Ethernet interface as the third interface, and another Token Ring as the fourth interface, the interfaces would be named **ethernet0**, **token0**, **ethernet1**, and **token1**.

  If one of the Ethernet cards is a 4-port card, the Ethernet names change to correspond to in which slot the card resides. However the Token Ring card names stay the same. For example, if slot 0 has a single port Ethernet card, the slot 1 has a 4-port card, and slot 2 has a Token Ring card, the interfaces would be named as follows:

  — For the single port card in slot 0, **ethernet0**.

  — For the 4-port card in slot 1, **ethernet1**, **ethernet2**, **ethernet3**, and **ethernet4**.

  — For the Token Ring card in slot 2, **token0**.

  You can abbreviate the *hardware_id* name with any significant letters, such as, **e0** for **ethernet0**, or **t0** for **token0**.

- *interface*—If you want to use names other than the default names, you can enter a name such as **dmz** or **perim** for each perimeter interface. Whichever name you pick, you will need to enter it repeatedly as you create your configuration, so a short name, such as **dmz**, will be easier to enter. However, if you want to, you can specify up to 48 characters in an interface name.
- *security_level*—A value such as **security40** or **security60**. You can choose any security level between 1 and 99 for a perimeter interface as long as it is not the same as the inside and outside interfaces. If you have four or more interfaces, it will be easier to code your configuration if you use the higher security level for the perimeter interface with the most hosts. When you access a higher security level interface from a lower security level interface, you use the **static** command.

  If you are configuring PIX Firewall for the first time, the default security levels for perimeter interfaces start with security10 for intf2 (the default name for the first perimeter interface), security15 for intf3, security20 for intf4, and security25 for intf5.

  When you access a lower security interface from a higher security level interface, you use the **nat** command. By using the higher security level, hosts on that interface can access the other perimeter interface and the outside interface using the **nat** command.

The **ip address** Command

Assign an **ip address** command to each interface in your PIX Firewall that connects to the network. For unused interfaces, PIX Firewall assigns 127.0.0.1 (the local host address) to each interface and a subnet mask of 255.255.255.255 that does not permit traffic to flow through the interface. The 127.0.0.1 address is the Internet address for the local host and is not used by any Internet site.

The format for the **ip address** command is:

```
ip address inside ip_address netmask
ip address outside ip_address netmask
```

Replace *ip_address* with the IP address you specify for the interface. The IP addresses that you assign must be unique for each interface—do not use an address you previously used for routers, hosts, or with any other PIX Firewall command, such as an IP address in the global pool or for a static.

Replace *netmask* with the network mask for the IP address; for example, 255.0.0.0 for a Class A address (those that begin with 1 to 127), use 255.255.0.0 for Class B addresses (those that begin with 128 to 191), and 255.255.255.0 for Class C addresses (those that begin with 192 and higher). Do not use 255.255.255.255 for an interface connected to the network because this will stop traffic on that interface.

If subnetting is in use, use the subnet in the mask; for example, 255.255.255.228.

Use the **show ip** command to view the commands you entered. If you make a mistake while entering a command, reenter the same command with new information.

An example **ip address** command is:

```
ip address inside 192.168.1.1 255.255.255.0
```

If you are using subnetting, enter a network mask applicable to the subnet. Refer to Appendix D, “Subnet Masking and Addressing” to ensure that the IP address you pick for each interface is correct for the subnet.

The **interface** Command

If you have Ethernet interfaces in the PIX Firewall, the default configuration provides **interface** commands for all interfaces.

Note Starting with version 5.0, all interfaces in a new configuration are shut down by default and need to be explicitly enabled for use.

Upgraded configurations from a previous PIX Firewall version are not affected by this new feature.

The format for this command is:

interface *hardware_id hardware_speed* [**shutdown**]

where:

- *hardware_id*—Either **ethernet***n* for Ethernet or **token***x* for Token Ring depending on how you specified the *hardware_id* in the **nameif** command.
- *hardware_speed*—If the interface is Token Ring, either **4mbps** or **16mbps** depending on the line speed of the Token Ring card. If the interface is Ethernet, and the PIX Firewall uses the Intel 10/100 interface type, use **auto**. If you purchased your PIX Firewall before November 1996, the Ethernet interfaces do not auto sense the interface speed. Refer to the **interface** command page in Chapter 6, "Command Reference," for how to specify the *hardware_speed*.
- **shutdown**—Disables use of the interface. When you first install PIX Firewall version 5.0, all interfaces have the **shutdown** option enabled. To enable use of the interface, recode the **interface** command without the **shutdown** option. For example, the starting configuration appears as follows for a four-interface PIX Firewall:

  ```
  interface ethernet0 auto shutdown
  interface ethernet1 auto shutdown
  interface ethernet2 auto shutdown
  interface ethernet3 auto shutdown
  ```

  For each interface you intend to operate, you need to reenter each command without the **shutdown** option. The following example enables the first three interfaces and leaves the last interface shutdown:

  ```
  interface ethernet0 auto
  interface ethernet1 auto
  interface ethernet2 auto
  ```

Use the **write terminal** command to view the configuration and locate the **interface** command information. If you make a mistake while entering a command, reenter the same command with new information.

Examples of the **interface** command are:

```
interface ethernet0 auto
interface token0 16mbps
```

Step 6 - Let Users Start Connections

As described in the section, "Step 5 - Identify Each Interface," the **nameif** command assigns a security level to each interface. For interfaces with a higher security level such as the inside interface, or a perimeter interface relative to the outside interface, use the **nat** and **global** commands to let users on the higher security interface access a lower security interface. For the opposite direction, from lower to higher, you use the **static** and **conduit** commands described in the section "Step 13 - Add Server Access."

As you enter the **nat** and **global** commands to let users start connections, you can use the **show nat** or **show global** commands to list the existing commands. If you make a mistake, remove the old command with the **no** form of the command, specifying all the options of the first command. This is where a terminal with cut and paste capability is useful. After you use **show global**, you can cut the old command, enter **no** and a space on the command line, paste the old line in, and press the **Enter** key to remove it.

PIX Firewall favors the use of NAT (Network Address Translation) for addressing of your network. When a PIX Firewall is first inserted in a network, keeping existing addressing may appear desirable, but imposes an extra layer of complexity in working with the PIX Firewall. Almost all of the PIX Firewall commands that work with IP addresses are affected by the use of NAT.

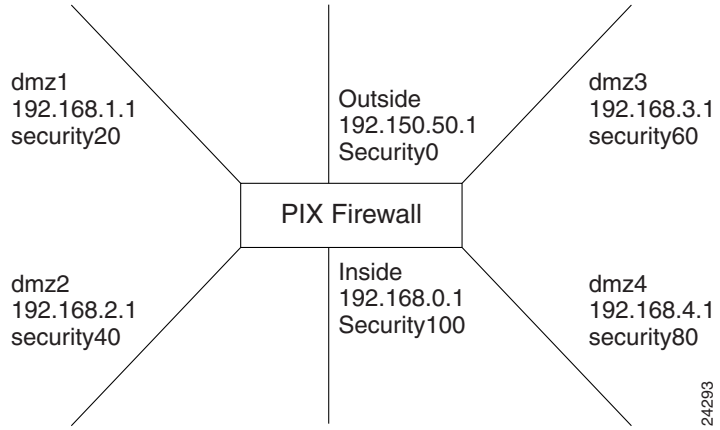
As you enter each command and debug it, you have to work with how your network addressing affects server access, creating global pools, authentication, routing, and starting connections. When you add in multiple interfaces, the complexity rises more. For this reason, Cisco recommends that you use PIX Firewall with NAT if possible. If you must disable NAT, use the **nat 0** command. Refer to the **nat** and **static** command pages, described in Chapter 6, "Command Reference," for a discussion of the implications of disabling NAT.

To let users on a higher security level interface start connections:

Step 1 Use the **show nameif** command to view the security level of each interface.

Step 2 Make a simple sketch of your network with each interface and its security level as shown in Figure 2-1.

Figure 2-1 Sketching Interfaces and Security Levels



Step 3 Add a **nat** command statement for each higher security level interface from which you want users to start connections to interfaces with lower security levels:

- To let inside users start connections on any lower security interface, use the **nat (inside) 1 0 0** command.

- To let dmz4 users start connections on any lower security interface such as dmz3, dmz2, dmz1, or the outside, use the **nat (dmz4) 1 0 0** command.
- To let dmz3 users start connections on any lower security interface such as dmz2, dmz1, or the outside, use the **nat (dmz3) 1 0 0** command.
- To let dmz2 users start connections on any lower security interface, such as dmz1 or outside, use the **nat (dmz2) 1 0 0** command.
- To let **dmz1** users start connections to the outside, use the **nat (dmz1) 1 0 0** command,

Instead of specifying "**0 0**," to let all hosts start connections, you can specify a host or a network address and mask.

For example, to let only host 192.168.2.42 start connections on the dmz2 interface, you could specify:

```
nat (dmz2) 1 192.168.2.42 255.255.255.255
```

The "1" after the interface specifier is the NAT ID. You can use one ID for all interfaces and the PIX Firewall sorts out which **nat** command statement pertains to which **global** command statement on which interface, or you can specify a unique NAT ID to limit access to specific interface. Remember that the **nat** command opens access to all lower security level interfaces so that if you want users on the inside to access the perimeter interfaces as well as the outside, then use one NAT ID for all interfaces. If you only want inside users to access the dmz1 interface but not the outside interface, use unique NAT IDs for each interface.

The NAT ID in the **nat** command must be the same NAT ID you use for the corresponding **global** command.

NAT ID **0** means to disable Network Address Translation.

Step 4 Add a **global** command statement for each lower security interface which you want users to have access to; for example, on the outside, dmz1, and dmz2. The **global** command creates a pool of addresses that translated connections pass through.

There must be enough global addresses to handle the number of users each interface may have trying to access the lower security interface. You can specify a single PAT (Port Address Translation) which permits up to 65,000 hosts to use a single IP address. PAT has some restrictions in its use such as it cannot support H.323 or caching nameserver use, so you may want to use it to augment a range of global addresses rather than using it as your sole global address.

For example:

```
global (outside) 1 192.150.50.9 netmask 255.255.255.0
global (outside) 1 192.150.50.10-192.150.50.20 netmask 255.255.255.0
```

The first **global** command statement specifies a single IP address, which the PIX Firewall interprets as a PAT. The PAT lets up to 65,535 hosts start connections to the outside. PIX Firewall permits one PAT global command statement for each interface The second **global** command statement augments the pool of global addresses on the outside interface. The PAT creates a pool of addresses used only when the addresses in the second **global** command statement are in use. This minimizes the exposure of the PAT in the event users need to use H.323 applications.

```
global (dmz1) 1 192.168.1.10-192.168.1.100 netmask 255.255.255.0
global (dmz2) 1 192.168.2.10-192.168.2.100 netmask 255.255.255.0
```

The **global** command statement for dmz1 lets users on the inside and dmz2 start connections on the dmz1 interface.

The **global** command statement for dmz2 lets users on the inside start connections on the dmz2 interface.

If you use network subnetting, specify the subnet mask with the **netmask** option. Refer to Appendix D, "Subnet Masking and Addressing" for more information on subnetting.

Step 7 - Create a Default Route

Use the **route** command to set a default route to the outside router. Use the **show route** command to view the command you entered. If needed, use the **no route** command to remove a **route** command. If the outside router is at address 192.150.50.3, you would use this command:

```
route outside 0 0 192.150.50.2 1
```

This command states that the default router is on the outside interface. The 0 0 information is an IP address of 0.0.0.0 and mask of 0.0.0.0, which the PIX Firewall associates with the default route. The **route** command could be read as "if I have a packet intended for IP address 0.0.0.0, send it to 192.150.50.2 instead." The "1" at the end is the number of hops that the router is from the PIX Firewall. Hops are routers, so 1 hop is the router nearest the PIX Firewall.

If the PIX Firewall has only two interfaces, you can specify a default route for the inside. Note that this exception only applies when the PIX Firewall has physically two interfaces. If a third interface is present in the firewall without a cable connection, you have to physically remove the card before you can use this exception. If there are only two interfaces, the default **route** command for the inside will eliminate having to add static **route** command statements for the networks connected to the inside router (if any).

Note If you are not sure how many interfaces are in the PIX Firewall, examine the configuration with the **write terminal** command and count the number of **interface** commands that appear. If there are 3 or 4, you must only have one default **route** command statement to the outside interface; otherwise, the PIX Firewall will experience routing problems that are difficult to diagnose.

Step 8 - Permit Ping Access

Enter the **conduit permit icmp any any** command in your configuration. This lets hosts on the inside ping outside hosts and hosts on the outside ping global addresses configured with the **conduit** command.

Step 9 - Store the Image in Flash Memory and Reboot

When you complete entering commands in the configuration, save it to Flash memory with the **write memory** command.

Then use the **reload** command to restart the configuration. After you enter the **reload** command, PIX Firewall prompts you to confirm that you want to continue. Enter **y** and the reboot occurs.

You are now done configuring the PIX Firewall. This configuration lets protected network users start connections, but prevents users on unprotected networks from attacking protected hosts.

Step 10 - Check the Configuration

Use the **write terminal** command to view your current configuration. Check the following before proceeding to ensure that your configuration is correct:

Step 1 Make sure that the each interface you intend to operate has the **shutdown** option disabled. Refer to the section "The interface Command" for more information.

Step 2 Make sure that the IP addresses you use in the **ip address**, **global**, **nat**, and **route** commands are unique. In addition, the **ip address** command IP address cannot be the same as a router or any hosts. Use the following commands to examine this information:

```
show ip address
show global
show nat
show route
```

Step 3 Use the **show route** command to make sure you have a default route command statement pointing to the outside router. A default **route** command follows:

```
route outside 0 0 ip_address_of_outside_router 1
```

Replace *ip_address_of_outside_router* with the IP address of the nearest router on the outside interface.

If you do not see this command in your configuration, add it now. A default **route** command is crucial to get other commands to work correctly. If you are testing the network before putting it into production, get a router and add it to the test network so that the PIX Firewall has a default route.

Step 4 Make sure that the **nat** and **global** command statements have the same NAT ID, as shown in the following example:

```
nat (dmz) 1 0 0
global (outside) 1 192.150.50.2 netmask 255.255.255.0
```

The number 1 after the interface name is the NAT ID.

Also, it is best to keep all the **nat** command statements and **global** command statements in the same NAT ID even if the **global** command statements refer to different interfaces, for example:

```
nat (inside) 1 0 0
nat (dmz1) 1 0 0
nat (dmz2) 1 0 0
global (outside) 1 192.150.50.2 netmask 255.255.255.0
global (outside) 1 192.150.50.10-192.150.50.200 netmask 255.255.255.0
global (dmz1) 1 192.168.1.20-192.168.1.200 netmask 255.255.255.0
```

The **nat** command statements let users on the inside, dmz1, and dmz2 interfaces start outside connections. The first **global** command statement creates a PAT address on the outside interface. The second **global** command statement creates a pool of IP addresses in the range of 192.150.50.30 to 192.150.50.200 on the outside interface.

The third **global** command statement creates a pool of IP addresses on the dmz1 interface. You can have one PAT global command statement per interface.

Step 5 Use the **show global** command to make sure that a range of global addresses starts from a low number and goes to a high number. In addition, it is good to leave a few addresses before the range for **static** command statements, hosts, or additional routers. In other words, instead of starting the global pool at an address such as 192.150.50.2, use 192.150.50.10 (assuming you have a full Class C address to use).

Step 6 If your ISP (Internet service provider) has only provided a few registered addresses, always have a PAT address at the end of the range. This expands your pool of addresses, if needed. Remember to give the PAT an address lower than the pool of global addresses. PIX Firewall uses global addresses starting from the highest numbered IP address and works down.

Step 7 If you are using subnetting, examine Appendix D, "Subnet Masking and Addressing," for more information on subnetting. Use the **show global** command to make sure that all addresses in the global pool are in the same subnet. For example, if you have a 255.255.255.192 subnet mask, the pool of global addresses could not contain addresses 192.150.50.60-192.150.50.100 because this would cross subnet boundaries.

Also make sure that the global pool contains correctly subnetted network addresses and broadcast addresses as explained in Appendix D, "Subnet Masking and Addressing." For example, with the 255.255.255.224 mask, specifying a global pool of 192.150.50.64-192.150.50.95 would not work because 192.150.50.64 is a network address and 192.150.50.95 is a broadcast address.

(a) Use the **show ip address** command to ensure that addresses on each interface are in the correct subnet for that interface. Each interface needs its own subnet. For example, if the outside interface has registered addresses 192.150.50.0 through 192.150.50.31 with a 255.255.255.224 subnet mask, the outside interface, outside router, any hosts on this interface, the global pool, and any addresses set aside for **static** command statements (explained in "Step 13 - Add Server Access") must all reside on addresses 192.150.50.1 through 192.150.50.30.

(b) If you are using subnetting, put the subnet value in the **ip address** and **global** command statements masks. For example, if you are using a .192 subnet mask, the **ip address** command would appear as:

```
ip address outside 192.150.50.1 255.255.255.192
```

The **global** command would appear as:

```
global (outside) 1 192.150.50.75-192.150.50.126 netmask 255.255.255.192
```

Step 8 Use the **show nat** command If you need to restrict IP addresses in **nat** command statements, do not overlap the groups. An example follows

```
nat (dmz1) 1 10.0.0.0 255.0.0.0
```

If you want only users on the 10.0.0.0 network to start connections, do not specify a second **nat** group with address 10.1.1.0 because this network would be included in 10.0.0.0.

Step 9 Use the **show ip address** command to check all IP addresses to be sure you have the correct addresses values for the devices.

Make sure all inside interface or perimeter interface hosts and routers have their default routes set to the respective PIX Firewall interface IP address. Refer to section "Step 3 - Configure Network Routing" for more information.

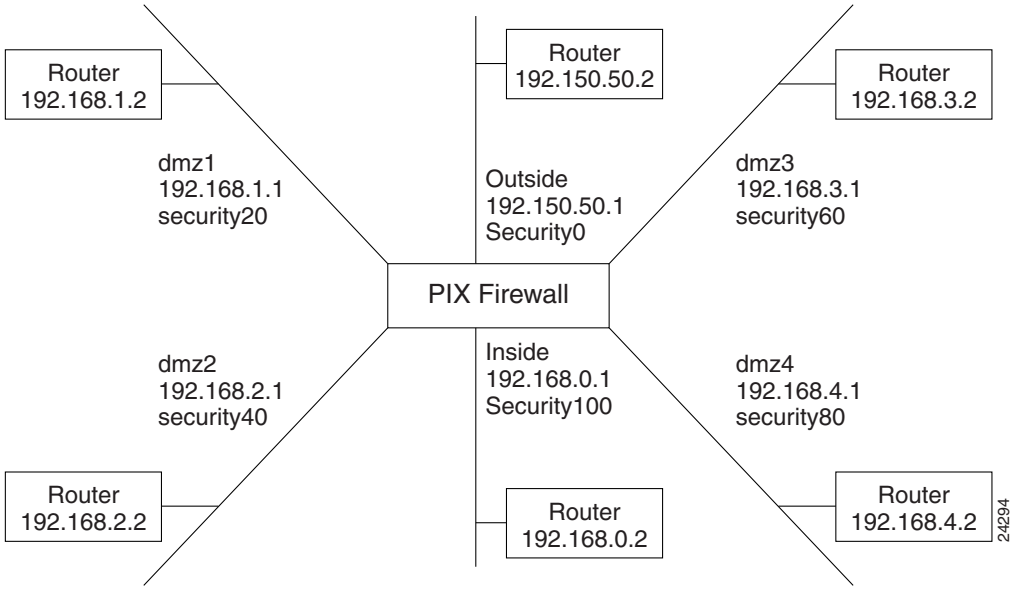
Step 11 - Test Network Connectivity

For the steps that follow, you will need access to the PIX Firewall console and to at least one host on both the internal and external networks.

Use the steps that follow to determine whether or not the firewall is functioning correctly in the network:

Step 1 **Sketch a diagram of your network**—With a sketch, it is much easier to methodically test the network with the PIX Firewall to be sure if everything works as expected as shown in Figure 2-2.

Figure 2-2 Sketch a Network with Interfaces and Routers



Step 2 **Start debugging commands**—Enter configuration mode and start the **debug icmp trace** command to monitor ping results through the PIX Firewall. In addition, start syslog logging with the **logging buffered debugging** command to check for denied connections or ping results. The **debug** messages display directly on the console session. You can view syslog messages with the **show logging** command.

Before using the **debug** command, use the **who** command to see if there are any Telnet sessions to the console. If the **debug** command finds a Telnet session, it automatically sends the **debug** output to the Telnet session instead of the console. This will cause the serial console session to seem as though no output is appearing when it is really going to the Telnet session.

Step 3 **Ping around the PIX Firewall**—Ping from the PIX Firewall to a host or router on each interface. Then go to a host or router on each interface and ping the PIX Firewall's interface. For the example, you would use these commands from the PIX Firewall:

```
ping inside 192.168.0.2
ping dmz1 192.168.1.2
ping dmz2 192.168.2.2
ping dmz3 192.168.3.2
ping dmz4 192.168.4.2
ping outside 192.150.50.2
```

Then ping the PIX Firewall interfaces from the hosts or routers with commands such as:

- Ping the PIX Firewall's outside interface with **ping 192.150.50.1**
- Ping the PIX Firewall's inside interface with **ping 192.168.0.1**
- Ping the PIX Firewall's dmz1 interface with **ping 192.168.1.1**
- Ping the PIX Firewall's dmz2 interface with **ping 192.168.2.1**
- Ping the PIX Firewall's dmz3 interface with **ping 192.168.3.1**
- Ping the PIX Firewall's dmz4 interface with **ping 192.168.4.1**

If the pings from the hosts or routers to the PIX Firewall interfaces are not successful, check the **debug** messages which should have displayed on the console. Successful ping debug messages appear as in this example:

```
ICMP echo request (len 32 id 1 seq 512) 192.150.50.42 > 192.150.50.1
ICMP echo reply (len 32 id 1 seq 256) 192.150.50.1 > 192.150.50.42
```

Both the request and reply statements should appear to show that the PIX Firewall and the host responded. If none of these messages appeared while pinging the interfaces, then there is a routing problem between the host or router and the PIX Firewall that caused the ping (ICMP) packets to never arrive at the PIX Firewall.

Also try the following to fix unsuccessful pings:

(a) Make sure you have a default **route** command statement for the outside interface. For example:

```
route outside 0 0 192.150.50.2 1
```

(b) Use the **show conduit** command to ensure that the **conduit permit icmp any any** command is in the configuration. Add this command if it is not present.

(c) Except for the outside interface, make sure that the host or router on each interface has the PIX Firewall as its default gateway. If so, set the host's default gateway to the router and set the router's default route to the PIX Firewall. Setting default routes in routers and hosts is explained in the section "Step 3 - Configure Network Routing."

(d) Check to see if there is a router between the host and the PIX Firewall. If so, make sure the default route on the router points to the PIX Firewall interface. If there is a hub between the host and the PIX Firewall, make sure that the hub does not have a routing module. If there is a routing module, configure its default route to point to the PIX Firewall.

(e) Go to the PIX Firewall and use the **show interface** command to ensure that the interface is functioning and that the cables are connected correctly. If the display contains "line protocol is up," then the cable type used is correct and connected to the firewall. If the display states that each interface "is up," then the interface is

ready for use. If both of these are true, check "packets input" and "packets output." If packets are being received and transmitted, the firewall is correctly configured and a cable is attached.

(f) Check that network cables are attached.

Ping through the PIX Firewall—Once you can ping the PIX Firewall's inside interface, try pinging through the PIX Firewall to a host on another interface, such as the outside. If there is not a host on the interface, ping the router. If the ping is not successful, check the debug messages on the PIX Firewall console to be sure both inbound and outbound pings were received. If you see the Inbound message without the Outbound, then the host or router is not responding. Check that the **nat** and **global** command statements are correct and that the host or router is on the same subnet as the outside interface. Successful ping debug messages appear as in this example:

```
Inbound ICMP echo reply (len 32 id 1 seq 256) 192.150.50.1 > 192.150.50.42
Outbound ICMP echo request (len 32 id 1 seq 512) 192.150.50.42 > 192.150.50.1
```

Step 4 Once you can ping successfully across interfaces of higher security levels to lower security levels, such as inside to outside, inside to dmz, or dmz2 to dmz1, add **static** and **conduit** command statements as described in the section "Step 13 - Add Server Access" so that you can ping from the lower security level interfaces to the higher security level interfaces.

Step 12 - Add Telnet Console Access

The serial console lets a single user configure the PIX Firewall, but many times this is not convenient for a site with more than one administrator. PIX Firewall lets you access the serial console via Telnet from hosts on any internal interface.

With PIX Firewall version 5.0 and IPSec configured, you can use Telnet to remotely administer the console of a PIX Firewall from the outside interface.

To configure Telnet console access:

Step 1 Use the PIX Firewall **telnet** command. For example, to let a host on the internal interface with an address of 192.168.1.2 access the PIX Firewall, enter:

```
telnet 192.168.1.2 255.255.255.255 inside
```

If IPSec is in place, you can let a host on the outside interface access the PIX Firewall console. Use a command such as:

```
telnet 209.165.200.225 255.255.255.224 outside
```

The **telnet** command does not have an interface identifier. The PIX Firewall compares the IP address you specify in the **telnet** command to those in the **ip address** command statements to ensure that the address you specify is on an internal interface (any interface except the outside interface).

Step 2 If required, set the duration for how long a Telnet session can be idle before PIX Firewall disconnects the session. The default duration, 5 minutes, is too short in most cases and should be increased until all pre-production testing and troubleshooting has been completed. Set a longer idle time duration as shown in the following example:

```
telnet timeout 15
```

Step 3 If you want to protect access to the console with an authentication server, you can use the **aaa authentication telnet console** command, which requires that you have a username and password on the authentication server. When you access the console, PIX Firewall

prompts you for these login credentials. If the authentication server is offline, you can still access the console by using the username **pix** and the password set with the **enable password** command.

Step 4 Save the commands in the configuration using the **write memory** command.

To test Telnet access:

Step 1 From the host, start a Telnet session to a PIX Firewall interface IP address. If you are using Windows 95 or Windows NT, click **Start>Run** to start a Telnet session. For example, if the inside interface IP address is 192.168.1.1, enter the following command:

```
telnet 192.168.1.1
```

Step 2 The PIX Firewall prompts you with a password:

```
PIX passwd:
```

Enter **cisco** and press the **Enter** key. You are then logged into the PIX Firewall.

The default password is **cisco**, which you can change with the **passwd** command.

You can enter any command on the Telnet console that you can set from the serial console, but if you reboot the PIX Firewall, you will need to log back into the PIX Firewall after it restarts.

Some Telnet applications such as the Windows 95 or Windows NT Telnet sessions may not support access to the PIX Firewall's command history feature used with the arrow keys. However, you can access the last entered commands by pressing Ctrl-P.

Step 3 Once you have Telnet access available, you may want to view ping information while debugging. You can view ping information from Telnet sessions with the **debug icmp trace** command. The Trace Channel feature also affects **debug** displays, which is explained in the section "Trace Channel Feature."

Messages for a successful ping appear as:

```
Inbound ICMP echo reply (len 32 id 1 seq 256) 192.150.50.1 > 192.150.50.42
Outbound ICMP echo request (len 32 id 1 seq 512) 192.150.50.42 > 192.150.50.1
```

Step 4 In addition, you can use the Telnet console session to view syslog messages:

(a) Start message displays with the **logging monitor 7** command. The "7" will cause all syslog message levels to display.

If you are using the PIX Firewall in production mode, you may wish to use the **logging buffered 7** command to store messages in a buffer that you can view with the **show logging** command, and clear the buffer for easier viewing with the **clear logging** command. To stop buffering messages, use the **no logging buffered** command.

You can also lower the number from **7** to a lesser value, such as **3,** to limit the number of messages that appear.

(b) If you entered the **logging monitor** command, then enter the **terminal monitor** command to cause the messages to display in your Telnet session. To disable message displays, use the **terminal no monitor** command.

Trace Channel Feature

The **debug icmp trace** and **debug sqlnet** commands send their output to the Trace Channel. The location of the Trace Channel depends on whether you have a simultaneous Telnet console session running at the same time as the console session, or if you are using only the PIX Firewall serial console:

- If you are only using the PIX Firewall serial console, all **debug** commands display on the serial console.
- If you have both a serial console session and a Telnet console session accessing the console, then no matter where you enter the **debug icmp trace** or the **debug sqlnet** commands, the output displays on the Telnet console session.
- If you have two or more Telnet console sessions, the first session is the Trace Channel. If that session closes, the serial console session become the Trace Channel. The next Telnet console session that accesses the console will then become the Trace Channel.
- The **debug packet** command only displays on the serial console. However, you can enable or disable this command from either the serial console or a Telnet console sessions.

The **debug** commands are shared between all Telnet and serial console sessions.

Note The downside of the Trace Channel feature is that if one administrator is using the serial console and another administrator starts a Telnet console session, the output from the **debug icmp trace** and **debug sqlnet** commands on the serial console will suddenly stop without warning. In addition, the administrator on the Telnet console session will suddenly be viewing **debug** command output, which may be unexpected. If you are using the serial console and **debug** command's output is not appearing, use the **who** command to see if a Telnet console session is running.

Step 13 - Add Server Access

By default, the PIX Firewall prevents all outside connections from accessing "inside" hosts or servers. Any server on a network that has a higher security level than the current interface requires a **static** and **conduit** command statement.

Note If you are using **nat 0**, refer to the **static** command page for information about how to handle server access in this environment.

For example, to let outside users access a dmz1 web server, you could have **static** and **conduit** command statements as follows:

```
static (dmz1,outside) 192.150.50.5 192.168.1.5 netmask 255.255.255.255
conduit permit tcp host 192.150.50.5 eq www any
```

In this example, the **static** command maps access to the dmz1 host 192.168.1.5 through a global address on the outside interface of 192.150.50.5. The **conduit** command lets any users on the outside access IP address 192.150.50.5 using a web browser on port 80 (**www**). In this example, the higher security level interface is dmz1 and the lower is the outside interface. On the outside interface, through the use of DNS, a company can map 192.150.50.5 to their web site address of www.caguana.com.

To help you code server access, use this rule for creating **static** command statements:

static (*high***,***low***)** *low high*

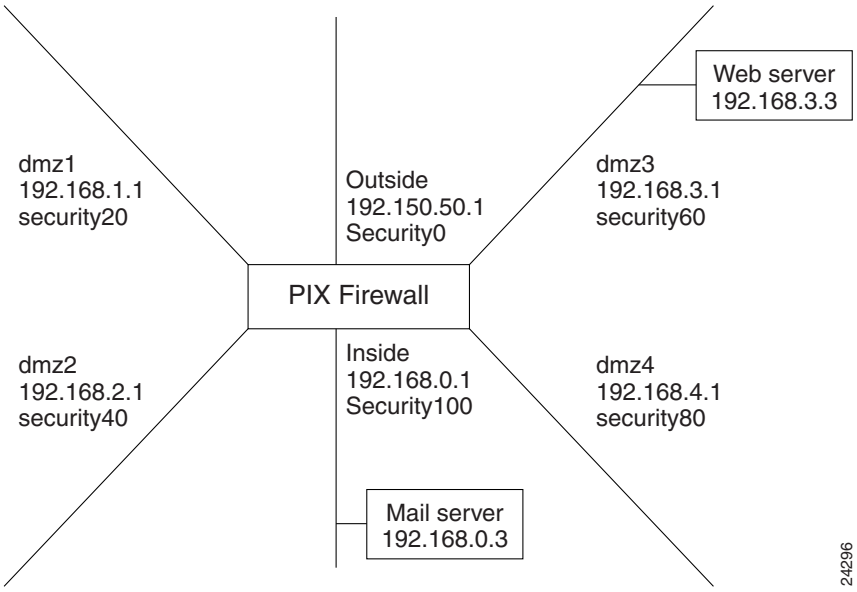
The idea is to present an IP address to users on one interface that gives them access to a host on another. You use the **static** command to let users on a lower security level interface access a server on a higher security level interface. You use the **nat** command to let users on a higher security level interface access a lower security level interface.

To create server access:

Step 1 View the security levels with the **show nameif** command.

Step 2 Sketch out a diagram of your network and label each interface with its security level and the IP addresses of the hosts you want to provide access to as shown in Figure 2-3.

Figure 2-3 Sketch a Network Diagram with Servers



From this scenario, you will need **static** command statements to let outside users access the dmz3 web server and for dmz1 and dmz2 users to access the web server. You will need a **nat** command statement to let inside and dmz4 users access the dmz3 web server.

For the mail server, you will need **static** command statements for access from the outside, dmz1, and dmz2, dmz3, and dmz4 interfaces.

Step 3 Provide access from the outside to the inside mail server with these commands:

```
static (inside,outside) 192.150.50.4 192.168.3.4 netmask 255.255.255.255
conduit permit tcp host 192.150.50.4 eq smtp any
```

These commands create a global address of 192.150.50.4 that PIX Firewall maps to the 192.168.3.4 mail server on the dmz2 interface. The **conduit** command statement permits any outside users to access the mail server at the SMTP port (25).

You will need to inform your DNS administrator to create an MX record for the global address (such as 192.150.50.4) so that mail is directed to the correct address.

We recommend that you not use the **any** keyword instead of specifying an IP address of a host that can access the static mapping in the **conduit** command statement. Using **any** lets any outside host access the static. In cases, such as for a server available for public access, using **any** is the only choice. However, if you can limit the number of users who have access to a server, you reduce the chance of intrusion.

PIX Firewall lets you specify a foreign IP address to protect access to the conduits. This is very important when there are multiple interfaces. If you set up a **conduit** command for the dmz2 interface to access the dmz1 interface, you would not want outside users to be able to access the **conduit** command. PIX Firewall handles this for you. It automatically determines which interfaces are mapped together with the **static** command statement.

Special Conduits

Two **conduit** command statements are required for establishing access to the following services: **discard**, **dns**, **echo**, **ident**, **pptp**, **rpc**, **sunrpc**, **syslog**, **tacacs-ds**, **talk**, and **time**. Each service, except for **pptp**, requires one **conduit** for TCP and one for UDP. For DNS, if you are only receiving zone updates, you only need a single **conduit** command statement for TCP.

The two **conduit** command statements for the PPTP transport protocol, which is a subset of the GRE protocol, are as shown in this example:

```
static (dmz2,outside) 192.150.50.5 192.168.1.5 netmask 255.255.255.255
conduit permit tcp host 192.150.50.5 eq 1723 any
conduit permit gre host 192.150.50.5 any
```

In this example, PPTP is being used to handle access to host 192.168.1.5 on the dmz2 interface from users on the outside. Outside users access the dmz2 host using global address 192.150.50.5. The first **conduit** command statement opens access for the PPTP protocol and gives access to any outside users. The second **conduit** permits access to GRE. If PPTP was not involved and GRE was, you could omit the first **conduit** command statement.

Step 4 Add the remaining **static** and **conduit** command statements:

- To let the dmz1 users access the mail server on the inside interface, create an IP address on the dmz1 interface that users can access that maps to the mail server:

```
static (inside,dmz1) 192.168.1.4 192.168.0.3 netmask 255.255.255.255
conduit permit tcp host 192.168.1.4 eq smtp any
```

- To let dmz2 users access the mail server:

```
static (inside,dmz2) 192.168.2.4 192.168.0.3 netmask 255.255.255.255
conduit permit tcp host 192.168.2.4 eq smtp any
```

- To let dmz3 users access the mail server:

```
static (inside,dmz3) 192.168.3.4 192.168.0.3 netmask 255.255.255.255
conduit permit tcp host 192.168.3.4 eq smtp any
```

- To let dmz4 users access the mail server:

```
static (inside,dmz2) 192.168.4.4 192.168.0.3 netmask 255.255.255.255
conduit permit tcp host 192.168.4.4 eq smtp any
```

These command statements create a global address on each interface to map to the inside mail server and then create a conduit so that users on each interface can access the mail server via the SMTP port (25).

Step 5 Let users know how to access the server. Users on the inside access the server at 192.168.3.4, users on the dmz1 interface access it at 192.168.1.4, and users on the dmz2 interface access it at 192.168.2.4.

To let users access the web server:

Step 1 Add command statements to let users on the various interfaces access the web server on dmz2.

- To let outside users access the web server on the dmz3 interface, create **static** and **conduit** command statements creating an IP address on the outside interface that maps to the web server on the dmz3 interface:

```
static (dmz3,outside) 192.150.50.3 192.168.3.3 netmask 255.255.255.255
conduit permit tcp host 192.150.50.3 eq www any
```

- To let dmz1 users access the web server:

```
static (dmz3,dmz1) 192.168.1.3 192.168.3.3 netmask 255.255.255.255
conduit permit tcp host 192.168.1.3 eq www any
```

- To let dmz2 users access the web server:

```
static (dmz3,dmz2) 192.168.2.3 192.168.3.3 netmask 255.255.255.255
conduit permit tcp host 192.168.2.3 eq www any
```

- To let dmz4 users access the web server, create **nat** and **global** command statements so that users on the dmz4, a higher security level interface than dmz3 start connections on dmz3:

```
nat (dmz4) 1 192.168.4.0 255.255.255.0
global (dmz3) 1 192.168.3.10-192.168.3.100 netmask 255.255.255.0
```

- To let inside users access the web server, add a **nat** command statement and the inside users can use the **global** command statement created for dmz4:

```
nat (inside) 1 192.168.0.0 255.255.255.0
```

The **static** and **conduit** command statements work the same way as described previously for the mail server, creating a global address through which users on the interface can access the web server. The **global** command adds a new dimension to server access. Because the inside interface is at a higher security level than the dmz2 interface, instead of using **static** and **conduit** command statements to permit access, you use **nat** and **global** command statements.

The **nat** command statement lets inside users start connections on any interface of a lower security level; therefore, they can access the dmz2 interface. The **global** command lets the inside users translate their connections to access the address of the web server on the dmz2 interface.

Step 2 Let users know what IP address to use to access the server. For users on the inside interface, they would access the web server at address 192.168.2.3, as would users on the same interface, dmz2. Users on dmz1 would access it at 192.168.1.3, and users on the outside would access it at 192.150.50.3.

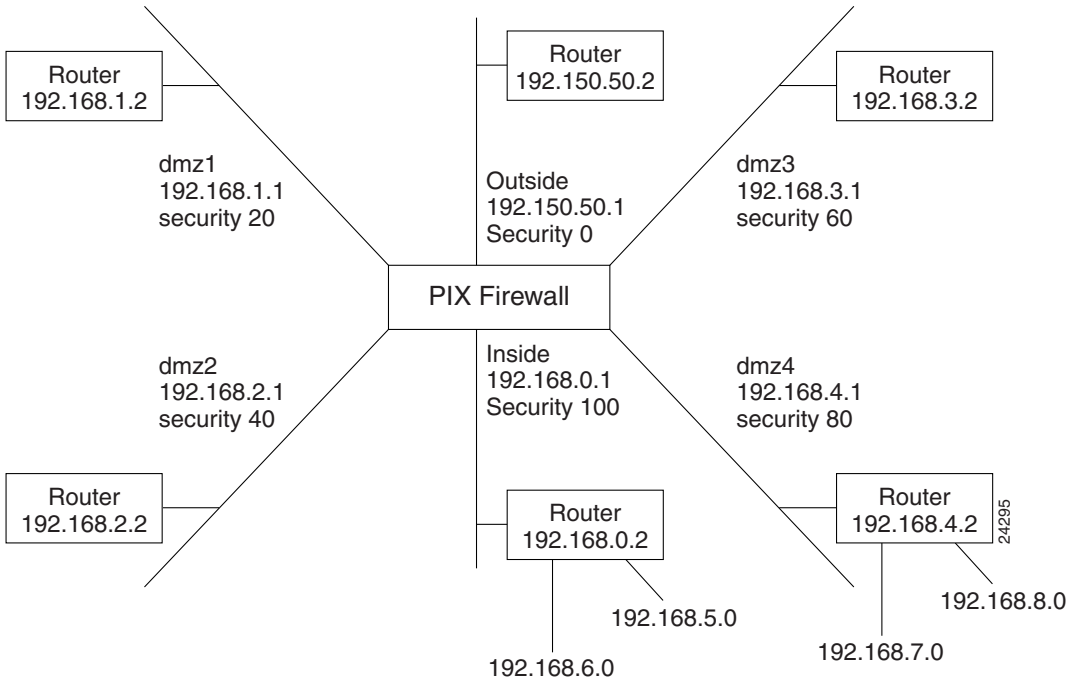
Step 14 - Add Static Routes

Specify a static **route** for each network connected to any router. Refer to the section "Step 7 - Create a Default Route" for information on default routes, and to the section "Step 3 - Configure Network Routing" for information on configuring routers and hosts for default routes.

To add static routes:

Step 1 Sketch out a diagram of your network as shown in Figure 2-4.

Figure 2-4 Sketch Network with Routes



Step 2 When you have three or more interfaces as shown in the diagram, only one default route is permitted:

```
route outside 0 0 192.150.50.2 1
```

This command statement sends all packets destined for the default route, IP address 0.0.0.0 (abbreviated as **0**, and **0** for the netmask), to the router 192.150.50.2. The "1" at the end of the command statement indicates that the router is the router closest to the PIX Firewall; that is, one hop away.

In addition, you must add static routes for the networks that connect to the inside router as follows:

```
route inside 192.168.5.0 255.255.255.0 192.168.0.2 1
route inside 192.168.6.0 255.255.255.0 192.168.0.2 1
```

These static **route** command statements can be read as "for packets intended for either network 192.168.5.0 or 192.168.6.0, ship them to the router at 192.168.0.2." The router decides which packet goes to which network. The PIX Firewall is not a router and cannot make these decisions.

The "1" at the end of the command statement specifies how many hops (routers) the router is from the PIX Firewall. Because it is the first router, you use 1.

Step 3 Add the static routes for the dmz4 interface:

```
route dmz4 192.168.7.0 255.255.255.0 192.168.4.2 1
route dmz4 192.168.8.0 255.255.255.0 192.168.4.2 1
```

These command statements direct packets intended to the 192.168.6.0 and 192.168.7.0 networks back through the router at 192.168.3.5.

Step 15 - Enable Syslog

The syslog message facility in the PIX Firewall is a useful means to view troubleshooting messages and to watch for network events such as attacks and service denials. You can view syslog messages either from the PIX Firewall console or from a syslog server that the PIX Firewall sends syslog messages to.

This section includes the following topics:

- Viewing Messages from the Console
- Viewing Messages from a Telnet Console Session
- Sending Messages to a Syslog Server
- Changing PFSS Parameters
- Recovering from Disk-full
- More on the logging Command
- Configuring a UNIX System for Syslog

Viewing Messages from the Console

To view messages from the PIX Firewall console:

Step 1 Use the **enable** command followed by the **configure terminal** command to get to configuration mode.

Step 2 Start storing messages in the PIX Firewall message buffer with the **logging** command:

```
logging buffered debugging
```

This command opens syslog up for all possible messages. The **debugging** setting is very useful for troubleshooting, but on a PIX Firewall in production, will generate too many messages to make troubleshooting viable. If you are testing a production mode PIX Firewall, substitute the **errors** keyword for the **debugging** keyword. This will reduce the messages to only those generated by logging levels 0, 1, 2, and 3. Refer to the *System Log Messages for the Cisco Secure PIX Firewall Version 5.0* guide for information about which messages display at each syslog level.

Step 3 Trigger some event in the PIX Firewall; for example, ping a host through the PIX Firewall. If your security policy permits pings, ensure that the ICMP **conduit** is in your configuration by using the **show conduit** command and checking for this command statement:

```
conduit permit icmp any any
```

If this command is not present, then add it.

Step 4 View the syslog messages with the **show logging** command. New messages append to the end of the display.

Step 5 To clear the messages in the buffer, use the **clear logging** command.

Step 6 When done, set the **logging buffered** command back to a minimal setting such as:

```
logging buffered alerts
```

This command will only store messages of levels 0 and 1.

Viewing Messages from a Telnet Console Session

To view syslog messages on a Telnet console session:

Step 1 Start Telnet from a host to an interface of the PIX Firewall. For example, to an internal interface:

```
telnet 192.168.1.3 255.255.255.255 inside
```

Step 2 The PIX Firewall prompts you for "PIX passwd:". Enter the Telnet password, which is **cisco** by default. (This password is set with the **passwd** command.)

Step 3 Use the **enable** command followed by the **configure terminal** command to get to configuration mode.

Step 4 Start message logging with the **logging monitor** command.

Step 5 Display messages directly to the Telnet session by entering the **terminal monitor** command.

Step 6 Use a host on an internal network to ping a host on the outside or start a web browser. These actions should create syslog events. The syslog messages then appear in the Telnet session window.

Step 7 To disable viewing syslog messages with Telnet, use these commands:

```
terminal no monitor
no logging monitor
```

The information in the remainder of this section describes additional information on the **logging** command and how to configure PIX Firewall to send messages to a syslog server.

Sending Messages to a Syslog Server

PIX Firewall can send syslog messages to either the PIX Firewall Syslog Server (PFSS) or to another syslog server such as those in UNIX or other operating systems.

In the event that all syslog servers are offline, PIX Firewall stores up to 100 messages in its memory. Subsequent messages that arrive overwrite the buffer starting from the first line.

If you have a Windows NT system, use of the PFSS gives you the additional benefit of reliability through receiving TCP event messages, receiving time stamped messages, and being able to monitor whether the server is up or down from the PIX Firewall. The PFSS is available without cost from Cisco Connection Online (CCO). Installation instructions for the PFSS are provided in the *Installation Guide for the Cisco Secure PIX Firewall Version 5.0*.

Note If your PIX Firewall is sending syslog messages via TCP to a PFSS and the Windows NT system's disk becomes full, the PIX Firewall will stop all new connections. If you are logging via UDP, the PIX Firewall does not check whether the disk becomes full.

Unless you need the certainty that every syslog message sent must be stored on the PFSS, and you can afford the possible network downtime to free the Windows NT disk space, only use UDP logging. If you use TCP logging, ensure that PFSS log files are backed up regularly to minimize the possibility of running out of disk space.

To send messages to a syslog server:

Step 1 Designate a host to receive the messages with the **logging host** command. For normal syslog operations to any syslog server, use the default message protocol, UDP, as shown in the following example:

```
logging host dmz1 192.168.1.5
```

If you want to use the reliable syslog feature of the PFSS whereby the PIX Firewall stops its traffic if the PFSS Windows NT disk becomes full or the system is unavailable, use the **tcp** option; for example:

logging host *interface address* **tcp/***port*

Replace *interface* with the interface on which the server exists, *address* with the IP address of the host, and *port* with the TCP port (if different than the default value of 1468).

You can see if PIX Firewall traffic has been disabled due to a PFSS disk-full condition with the **show logging** command and look for the "disabled" keyword in the display.

Only one UDP or TCP command statement is permitted for a server. A subsequent command statement overrides the previous one. Use the **write terminal** command to view the **logging host** command statement in the configuration. In the configuration, the UDP protocol appears as "17" and TCP as "6."

Step 2 Set the logging level with the **logging trap** command; for example:

```
logging trap debugging
```

Cisco recommends that you use the **debugging** level during initial setup and during testing. Thereafter, set the level from **debugging** to **errors** for production use.

Step 3 If needed, set the **logging facility** command to a value other than its default of 20. Most UNIX systems expect the messages to arrive at facility 20, which receives the messages in the local4 receiving mechanism, described in the section "Configuring a UNIX System for Syslog."

Step 4 Start sending messages with the **logging on** command. To disable sending messages, use the **no logging on** command.

Step 5 If you want to send time stamped messages to the PFSS, use the **clock set** command to set the PIX Firewall system clock and the **logging timestamp** command to enable time stamping. For example:

```
clock set 14:25:00 apr 1 2000
logging timestamp
```

In this example, the clock is set to the current time of 2:25 pm on April 1, 1999, and time stamping is enabled. To disable time-stamp logging, use the **no logging timestamp** command.

Step 6 If you want to stop sending a message to the syslog server, use the **no logging message** *syslog_id* command. Replace *syslog_id* with a syslog message ID, which you can view in the *System Log Messages for the Cisco Secure PIX Firewall Version 5.0*. You can access version 5.0 documentation online at:

http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_v50/index.htm

For example, to block the following message:

```
%PIX-6-305002: Translation built for gaddr IP_addr to IP_addr
```

Use this command to stop the message from being sent to the syslog server:

```
no logging message 305002
```

If you want to let the message resume being sent, use the following command:

```
logging message 305002
```

You can view disabled messages with the following command:

```
show logging disabled
no logging message 305002
```

You can re-enable all previously blocked messages with the following command:

```
clear logging disabled
```

Note The **no logging message** command cannot block the "`%PIX-6-199002: PIX startup completed. Beginning operation.`" syslog message.

Changing PFSS Parameters

You can change PFSS parameters at the Windows NT system using the **Start>Settings>Control Panel>Services** feature.

All PFSS parameter values can be viewed by examining the pfss.log file, which PFSS creates in the same directory as the PFSS log files.

The PFSS starts immediately after installation. You can use the **Services** control panel to enter new parameters, pause the service and then resume the service, or to stop and start the service. Choose one or more parameters from the following:

- **-d** *%_disk_full*—The maximum percentage of how full the Windows NT system disk can become before PFSS causes the PIX Firewall to stop transmissions. This is an integer value in the range of 1 to 100. The default is 90.
- **-t** *tcp_port*—The port that the Windows NT system uses to listen for TCP syslog messages, the default is 1468. If you specify another port, it must be in the range of 1024 to 65535.
- **-u** *udp_port*—The port that the Windows NT system uses to listen for UDP syslog messages, the default is 514. If you specify Another port, it must be in the range of 1024 to 65535.
- **-e** *disk_empty_watch_timer*—The duration in seconds that PFSS waits between checks to see if the disk partition is still empty. The default is 5 seconds, the range is any number greater than zero.
- **-f** *disk_full_watch_timer*—The duration in seconds that PFSS waits between checks to see if the disk partition is still full. The default is 3 seconds, the range is any number greater than zero.

For example, to set %_disk_full to 35 percent and the disk-full timer to 10 seconds:

Step 1 Open the Services control panel.

Step 2 Click the **PIX Firewall Syslog Server** service.

Step 3 In the Startup Parameters edit box, type **`-d 35 -f 10`**.

Step 4 Click **Start**. Pressing the **Enter** key closes the Services control panel and does not change the parameters.

PFSS stores syslog messages in one of seven files: monday.log, tuesday.log, wednesday.log, thursday.log, friday.log, saturday.log, sunday.log (according to the day of the week). If a week has already passed since the last log file was created, it will rename the old log file to *weekday.mmddyy* where *weekday* is the current day, *mm* is the month, *dd* is the day, and *yy* is the year; for example, monday.103099.

Note PFSS truncates syslog messages longer than 512 characters in length.

Recovering from Disk-full

If you have specified that the PIX Firewall send syslog messages via TCP, you may encounter the possibility that the Windows NT disk will become full and the PIX Firewall unit will stop its traffic. If the Windows NT file system is full, the Windows NT system beeps and the PFSS disables all TCP connections from the PIX Firewall unit(s) by closing its TCP listen socket.

The PIX Firewall tries to re-connect to the PFSS five times, and during the retry, it stops all new connections through the PIX Firewall. You then need to back up all the log files to another disk or across the network. (While PFSS is receiving messages, the log files must reside on the local disk.)

To recover from the disk-full condition:

Step 1 Back up the files on the Windows NT system.

Step 2 On the PIX Firewall, check that syslog is disabled with the **show logging** command. If the syslog server has disabled the connection, the display contains the “disable” keyword.

Step 3 Disable logging to the PFSS with the **no logging host** command; for example:

```
no logging host dmz1 10.1.1.2
```

Step 4 Restart logging with the **logging host** command; for example:

```
logging host dmz1 10.1.1.2 tcp/1468
```

Step 5 Check that the server is now enabled with the **show logging** command. The “disabled” keyword should no longer be visible.

More on the **logging** Command

The **logging facility** and **logging level** commands configure the facility and level of syslog messages. Because network devices share the eight facilities, the **logging facility** command lets you set the facility marked on all messages. Messages are sent to the syslog host over UDP. The **logging on** command starts sending messages. Use the **logging host** command to specify which systems receive the messages.

You can us the **show logging** command to view previously sent messages.

The PIX Firewall generates syslog messages for system events, such as security alerts and resource depletion. Syslog messages may be used to create email alerts and log files, or displayed on the console of a designated host using UNIX syslog conventions.

A PC WinSock version of **syslogd** will also work.

The PIX Firewall sends syslog messages to document the following events:

- Security—Dropped UDP packets and denied TCP connections.
- Resources—Notification of connection and translation slot depletion.
- System—Console and Telnet logins and logouts, and when the PIX Firewall reboots.
- Accounting—Bytes transferred per connection.

Logging is enabled by configuring the PIX Firewall with the IP address of the log host.

Syslog Facility and Level

The **logging facility** and **logging trap** commands let you specify the syslog facility and level for how messages are sent to the syslog host.

The facility consists of eight facilities LOCAL0(16) through LOCAL7(23); the default is LOCAL4(20). Hosts file the messages based on the facility number in the message.

The level specifies the types of messages sent to the syslog host. Setting the level to **3**, the default value, for example, allows messages with levels 0, 1, 2, and 3 to display. The default is **3**.

Table 2-2 lists syslog message levels.

Table 2-2 Syslog Message Levels

Use Level Number:	Or Use This Name:	For This Type of Message:
0	**emergencies**	System unusable messages
1	**alerts**	Take immediate action
2	**critical**	Critical condition
3	**errors**	Error message
4	**warnings**	Warning message
5	**notification**	Normal but significant condition
6	**informational**	Information message
7	**debugging**	Debug messages and log FTP commands and WWW URLs. For more information about logging FTP commands and URLs, refer to “FTP and URL Logging” in Chapter 3, “Advanced Configurations.”

Configuring a UNIX System for Syslog

After you have configured PIX Firewall to send syslog messages, configure either a PC or UNIX host to receive the messages. This section describes how to configure a UNIX host to receive syslog messages.

To configure a UNIX system to accept syslog messages:

Step 1 Use the PIX Firewall **logging host** command to configure the PIX Firewall to send syslog messages to the UNIX host's IP address.

Step 2 Log into the UNIX system as root (superuser) and execute the following commands:

```
# mkdir /var/log/pix
# touch /var/log/pix/pixfirewall
```

Step 3 While still logged in as root, edit the /etc/syslog.conf file with a UNIX editor and add a single selector and action pair for local4.error which will receive all the PIX Firewall syslog messages:

```
# PIX Firewall syslog messages
local4.error    /var/log/pix/pixfirewall
```

This configuration directs the PIX Firewall syslog message to the specified file. Alternatively, if you want the message sent to the logging host console or emailed to a system administrator, refer to the UNIX **syslog.conf**(4) manual page.

Note The UNIX log file can grow to several megabytes per day when monitoring a busy PIX Firewall.

Entries in /etc/syslog.conf must follow these rules:

(a) Comments, which start with the pound (#) character, are only allowed on separate lines.

(b) Separate the selector and action pairs with a tab character. Blanks are not acceptable.

(c) Ensure that there are no trailing spaces after the filenames.

Step 4 Inform the syslog server program on the UNIX system to reread the syslog.conf file by sending it a HUP (hang up) signal with the following command:

```
# kill -1 `cat /etc/syslog.pid`
```

This command lists the syslog process ID. This number may vary by system.

Step 16 - Create Access Lists

PIX Firewall provides the **outbound** and **apply** commands that you can use to limit internal users access to services on external interfaces. Use these commands to limit access for users who are on a higher security level interface from accessing a lower security level interface; for example, from the inside to the outside, from the inside to a perimeter interface, or between perimeter interfaces. These commands follow the direction of the **nat** command—also from a higher security level interface to a lower security level interface.

The **outbound** and **apply** commands' use is very interwoven. Depending on how you set the **apply** command, you use the **outbound** command to specify the details. The **apply** command lets you specify the interface you want to protect and how you are using the access list—to limit service access to your internal users (with the **outgoing_src** option, or to limit internal users access to a specific site (with the **outgoing_dest** option).

The **outbound** command specifies whether you are permitting or denying access, the affected IP addresses, and the port number or numbers. To coordinate the **outbound** and **apply** command statements, there is an identification number on both commands called the "list ID." The list ID is also used to order groups of commands so as to determine which group is processed first. This number is independent of the **nat** and **global** commands identification numbers—you can use the same number or another. Cisco recommends coding list IDs with gaps in the range to permit future additions, such as 10, 20, 30, or 100, 200, 300. Just be sure to use the same list ID on the **apply** command statement as on the **outbound** command for the same group. For example:

```
outbound 10 deny 0 0 www tcp
outbound 10 permit 192.168.1.2 255.255.255.255 www tcp
apply (dmz1) 10 outgoing_src
```

In addition, the order in which you specify the **outbound** commands determines how PIX Firewall evaluates them. The **outbound** command statements are ordered first by denies, then permits, and then by the list ID. Then there is the **except** option to this command, which has its own set of rules that are best viewed on the **outbound/apply** command page in Chapter 6, "Command Reference."

There are a few caveats with the **outbound** command. With the **outgoing_src** option to the **apply** command, you can only specify the internal hosts that are affected, not where you want them to go. Use the **outgoing_src** option to regulate access to services (ports) and protocols.

With the **outgoing_dest** option, you can only specify which host you do not want users to access, but not limit specific users access to the host. Use the **outgoing_dest** option to regulate access to a host.

Before creating an access list:

Step 1 Use the **show nameif** command to view the security levels of each interface.

Step 2 Use a **nat** command statement to let the users on the higher security level interface start connections on lower security level interfaces. For example, use **nat (inside) 1 0 0** to let inside users start connections, use **nat (dmz1) 1 0 0** for dmz1 users, or **nat (dmz2) 1 0 0** to let dmz2 users start connections.

To limit users access to a service:

Step 1 Create a blanket deny statement to limit higher security level users from accessing whatever service you are limiting. For example, to limit users on the 192.168.1.0 network on the dmz1 interface from using chat services on the outside, use the following command:

```
outbound 10 deny 192.168.1.0 255.255.255.0 irc tcp
```

Step 2 If required, permit access to those users who require access to this service; for example, to researchers who need to use chat in their work:

```
outbound 10 permit 192.168.1.42 255.255.255.255 irc tcp
```

Step 3 Then add the **apply** command statement to determine how you want to use the outbound list. In this example, because you are blocking every user at the source, use the **outgoing_src** option of the **apply** command:

```
apply (dmz1) 10 outgoing_src
```

To limit users access to a host:

Step 1 In this example, you want to keep dmz1 users from accessing a specific web site at 192.150.50.42 with objectionable material:

```
outbound 20 deny 192.150.50.42 255.255.255.255 www tcp
```

Step 2 Add the **apply** command statement:

```
apply (dmz1) 20 outgoing_dest
```

Step 17 - Add AAA User Authentication

User authentication and authorization starts with your security policy and the respective inside RADIUS or TACACS+ server that you have.

Authentication verifies that a user is who they say they are. Authorization determines what services a user can use to access a host.

From the configuration on this server you need to determine which users can access the network, which services they can use, and what hosts they can access. Once you have this information, you can configure the PIX Firewall to either enable or disable authentication or authorization.

In addition, you can also configure the firewall to permit users access to specific hosts or services. However, if you configure the firewall to this degree, you risk the information being different between the authentication server and the firewall. After you enable authentication and authorization, the PIX Firewall provides credential prompts to inbound or outbound users for FTP, Telnet, or HTTP (Web) access. The actual decision about who can access the system and with what services is handled by the authentication and authorization servers.

To provide user authentication and authorization:

Step 1 For inbound authentication, create the **static** and **conduit** command statements required to permit outside hosts to access servers on the inside network. This is described in "Step 13 - Add Server Access."

Step 2 If the external network connects to the Internet, create a global address pool of registered IP addresses, or if the network connects to an intranet, a pool of those addresses with the **global** command. Then specify which inside hosts can start outbound connections with the **nat** command and with the access control lists features found in the **outbound** and **apply** commands.

Step 3 Specify which server handles authentication or authorization with the **aaa-server** command. RADIUS can provide authentication but not authorization. Create a unique server group name. For example:

```
aaa-server AuthInbound protocol tacacs+
aaa-server AuthInbound (inside) host 10.1.1.1 TheUauthKey
aaa-server AuthOutbound protocol radius
aaa-server AuthOutbound (inside) host 10.1.1.2 TheUauthKey
```

The first command statement creates the AuthInbound authentication group using TACACS+ authentication. The second command statement states that the AuthInbound server is on the inside interface, that its IP address is 10.1.1.1, and the encryption key is "TheUauthKey."

The third command statement creates the AuthOutbound authentication group using RADIUS authentication. The fourth command statement states that the AuthOutbound server is on the inside interface, that its IP address is 10.1.1.2, and the encryption key is "TheUauthKey."

Step 4 Enable authentication with the **aaa authentication** command. It is best to use this command only to enable authentication with one or both of the following commands:

```
aaa authentication any outbound 0 0 0 0 AuthOutbound
aaa authentication any inbound 0 0 0 0 AuthInbound
```

The AuthInbound and AuthOutbound groups are those you specified with the **aaa-server** command.

Step 5 Enable authorization with the **aaa authorization** command. Even though this command lets you specify which services and inside hosts an authorized user can access, it is best to not set it here and only use this command to enable authorization. The authorization server should make the decision. Use one or both of the following commands:

```
aaa authorization any outbound 0 0 0 0 AtacacsGroup
aaa authorization any inbound 0 0 0 0 AuthInbound
```

You can specify port ranges for the **aaa authorization** command in the following format:

aaa authorization *service*|[*protocol*/*port*[-*port*]] **outbound 0 0 0 0** *group_tag*

where:

- *service*—is the service that PIX Firewall listens for AAA connections. Possible values are **any**, **http**, **ftp**, or **telnet**.
- *protocol*—is the protocol to authorize access to. Possible values are **udp**, **tcp**, or **icmp**.
- *port*—a port value or range to authorize users access to.
- *group_tag*—the group tag created with the **aaa-server** command.

Step 18 - Recheck the Configuration

When you have completed your configuration, check it carefully as described in the following steps and tips:

Step 1 If you are using the PIX Firewall Syslog Server (PFSS) and traffic through the PIX Firewall has stopped, first check the Windows NT system where the PFSS is installed and free the disk space if it is full. Once the disk space is freed, the PIX Firewall should restart sending traffic.

Step 2 Check that the interface addresses, global and NAT addresses, and route addresses are unique. All interfaces must be defined, have valid addresses, and appropriate subnet masks.

Step 3 If you have more than two interfaces, check the **nameif** command for the security level.

Step 4 If you are establishing access from a higher security level interface to a lower security interface, use the **nat** and **global** commands:

(a) Make sure that the NAT ID used in the **nat** command is the same NAT ID used in the **global** command.

(b) For the **global** command statement, ensure that you have enough global addresses for users in the network.

(c) Check the IP addresses to be sure they are correctly entered. Ensure that the **nat** command statement addresses do not overlap each other, or that the PAT address does not overlap the addresses in the global pool.

(d) Ensure that the global pool contains enough addresses for the number of clients on the interface to which it applies. If PAT is in use, ensure that it is configured with the same **nat** command statement identifier as the main pool of global addresses.

(e) If you have a global pool and if it is not on the same subnet as the router outside, the outside router *must* have a static route pointing back towards the outside interface of the PIX Firewall.

(f) If you use subnetting, be sure to specify a subnet mask with the **global** command and be sure that the addresses you specify are correct for the subnet mask range. Refer to Appendix D, "Subnet Masking and Addressing" for more information about subnet mask ranges.

Step 5 If you are establishing access from a lower security interface to a higher security interface, use the **static** and **conduit** commands:

(a) For server access, make sure that you have a **conduit** command for every **static** command you specify.

(b) Code **conduit** commands as tightly as possible. For example, specify which network can access the **conduit** and specify the exact port for which you permit access.

(c) Make sure that the global address in the **static** command is the same in the **conduit** command. For example if users on the dmz1 interface need to access a server on the dmz2 interface (dmz2 has a higher security level than dmz1), use commands similar to this example:

```
static (dmz2,dmz1) 10.1.1.2 192.168.1.2 netmask 255.255.255.255
conduit permit tcp host 10.1.1.2 eq smtp 10.1.1.0 255.255.255.0
```

In this example, the **static** command statement maps the 192.168.1.2 mail server on the dmz2 interface so that users on the dmz1 interface can access the server as 10.1.1.2. The **conduit** command statement specifies that only users on the 10.1.1.0 network can access the server via the SMTP port (25).

(d) Check that each **static** and **conduit** command statement pair has the correct addresses.

(e) Check that two **conduit** command statements are entered for establishing access to the following services: **discard**, **dns**, **echo**, **ident**, **pptp**, **rpc**, **sunrpc**, **syslog**, **tacacs-ds**, **talk**, and **time**. Each service, except for **pptp**, requires one **conduit** command statement for TCP and one for UDP. For DNS, if you are only receiving zone updates, you only need a single **conduit** command statement for TCP. Refer to the section "Step 13 - Add Server Access" for an example of two **conduit** command statements for the PPTP protocol.

Step 6 If an **outbound** list exists, ensure that the **apply** command statement is correct and that the list ID matches between the **outbound** and **apply** command statements.

Step 7 Ensure that the **route** command statements point to routers on appropriate interfaces. Ping these routers from the PIX Firewall to make sure they exist.

Step 8 Ensure that there is only one default **route** command statement to the outside interface.

Step 9 When you ping from an internal or external host during testing, use the **debug icmp trace** command to ensure that traffic is moving through the firewall correctly. Before using the **debug** command, use the **who** command to see if there are any Telnet sessions to the console. If the **debug** command finds a Telnet session, it automatically sends the **debug** output to the Telnet session instead of the console. This will cause the serial console session to seem as though no output is appearing when it is really going to the Telnet session.

Step 10 Consult with your ISP (Internet service provider) to make sure that all addresses used in **global** command statements are routed to your outside router before configuring the PIX Firewall with global addresses.

Step 11 If you use the same IP address range on all interfaces, IP addresses on the inside and outside (and perimeter) interfaces must be on different subnets.

Additional tips to consider are as follows:

- Ethernet network interface cards support both half and full duplex transmissions. However, the 3Com 10/100 card on earlier PIX Firewall units does not support 100 Mbps full duplex or the **100full** option to the **interface** command. These interfaces also report "line protocol down" with the **show interface** command.
- Use the **timeout** command to decrease the **xlate** and **conn** timers, if you see these syslog messages:

  ```
  %PIX-3-305005: No translation group found for packet
  %PIX-3-305006: xlate_type translation creation failed for packet
  ```

  When the messages display, the contents of the *packet* displays as text. The *xlate_type* can be either static, portmapped, or regular. Portmapped refers to a PAT global.

- If you have a router on an interface, the hosts on the other side of the router need a default gateway pointing to the router and the router needs a default gateway pointing to the PIX Firewall's respective interface.
- If you have two interfaces, you can use two default routes. If you have three or more interfaces, you can have only one default route to the outside router.
- Use the **write memory** command often to save your configuration to Flash memory.
- Use the **write memory** and **reload** commands after changing **alias**, **conduit**, **global**, **nat**, or **static** commands.
- Use the **no failover** command to disable failover if it is not in use.
- Make sure the MTU is no more than 1500 bytes for Ethernet, or 8192 for either Token Ring or FDDI.