CHAPTER 2

Configuring the PIX Firewall

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

Note If you are using a PIX Firewall unit that contains a diskette drive, you must use a "Boothelper" diskette to download the PIX Firewall image with TFTP. If your site has a Cisco router, the use of TFTP is similar to the way you download Cisco IOS software to your router.

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 Inbound Server Access
Step 3—Configure Network Routing	Step 7—Create a Default Route	Step 14—Add Outbound Access Lists
Step 4—Start Configuring PIX Firewall	Step 8—Permit Ping Access	Step 15—Add Static Routes
	Step 9—Store the Image in Flash Memory and Reboot	Step 16—Enable Syslog
	Step 10—Check the Configuration	Step 17—Add AAA User Authentication
	Step 11—Test Network Connectivity	Step 18—Recheck the Configuration

Also view Chapter 3, "Advanced Configurations," for information on configuring optional and advanced features.

For IPSec configuration information, refer to the *IPSec User Guide for the Cisco Secure PIX Firewall Version 5.3*.

Acronyms in this chapter are defined in Appendix B, “Acronyms and Abbreviations.” All commands shown in this chapter are explained fully in Chapter 5, “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.”

PIX Firewall displays a warning message if the configuration file (stored in Flash memory) is newer than the PIX Firewall software version currently being loaded. This message warns you of the possibility of unrecognized commands in the configuration file. For example, if you install version 5.2 software when the current version is 5.3, the following message appears at startup:

```
Configuration Compatibility Warning:
  The config is from version 5.3(1).
  but the image is version 5.2(1).
```

In the message, “config” is the version in Flash memory and “image” is the version you are installing.

Step 1—Get a Console Terminal

If the computer you are connecting to runs Windows, 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.

Follow these steps 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, first wait 60 seconds. 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:

- Get a TFTP Server
- Download the Latest Software from the Web
- Download the Latest Software with FTP
- Obtain the Boothelper Binary Image
- Use Boothelper to Download an Image

If you have a Cisco Connection Online (CCO) login, you can obtain software from the following site:

http://www.cisco.com/cgi-bin/tablebuild.pl/pix

The software available at this site includes the following items:

- **bh5***nn***.bin**—Lets you create a “Boothelper” installation diskette required to download version 5.3 PIX Firewall software from a TFTP server.
- **pix52***n***.bin**—The latest software image. Place this image in the TFTP directory so it can be downloaded to the PIX Firewall unit.
- **pfss5***nn***.exe**—Contains the PIX Firewall Syslog Server (PFSS), which installs on a Windows NT Server so that it can receive syslog messages from the PIX Firewall and store 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.
- **pfm432f.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 and must not be used with the PFSS. Version 4.3(2)f or later of the PFM accepts PIX Firewall versions 4.3, 4.4, 5.0, 5.1, 5.2, 5.3, and later, but it has not been upgraded with new command options in the current version. Refer to the *Release Notes for the PIX Firewall Manager Version 4.3(2)f* for more information on how to install and use this feature.

- **psw5*nn*.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, 5.1, 5.2, 5.3, and later. Refer to the *Installation Guide for the Cisco Secure PIX Firewall Version 5.3* for how to install the Setup Wizard.
- **rawrite.exe**—A program you use to create a Boothelper diskette for the PIX Firewall.

Get a TFTP Server

You must have a TFTP server to install the PIX Firewall software. If your computer runs the Windows operating system and you have a CCO login, you can download a TFTP server from Cisco from the Web or by FTP.

You can download the server from the Web at the following site:

http://www.cisco.com/cgi-bin/tablebuild.pl/tftp

Follow these steps to download the server by 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 top level software directory. Then enter **cd tftp** to access the TFTP software directory. Use the **ls** command to view the directory contents.

Step 4 Use the **get** command to copy the TFTP executable file to your directory.

The file you download is a self-extracting archive that you can use with Windows 95, Windows 98, or Windows NT version 4.0. Once the file is stored on your Windows system, double-click it to start the setup program. Then follow the prompts that appear to install the server on your system.

The UNIX, Solaris, and LINUX operating systems contain a TFTP server.

Note Under no circumstances must you ever download a PIX Firewall image earlier than version 4.4 with TFTP. Doing so will corrupt the PIX Firewall Flash memory unit and require special recovery methods that must be obtained from customer support.

Use the following steps to download an image over TFTP using the **monitor** command:

Step 1 Immediately after you power on the PIX Firewall and the startup messages appear, send a BREAK character or press the Esc (Escape) key.

The monitor> prompt appears.

Step 2 If desired, enter a question mark (**?**) to list the available commands.

Step 3 Use the **address** command to specify the IP address of the PIX Firewall unit's interface on which the TFTP server resides.

Step 4 Use the **server** command to specify the IP address of the host running the TFTP server.

Step 5 Use the **file** command to specify the filename of the PIX Firewall image. In UNIX, the file needs to be world readable for the TFTP server to access it.

Step 6 If needed, enter the **gateway** command to specify the IP address of a router gateway through which the server is accessible.

Step 7 If needed, use the **ping** command to verify accessibility. Use the **interface** command to specify which interface the ping traffic should use. If the PIX Firewall has only two interfaces, the **monitor** command defaults to the inside interface.If this command fails, fix access to the server before continuing.

Step 8 Use the **tftp** command to start the download.

An example follows:

```
Rebooting....
PIX BIOS (4.0) #47: Sat May 8 10:09:47 PDT 1999
Platform PIX-520
Flash=AT29C040A @ 0x300

Use BREAK or ESC to interrupt flash boot.
Use SPACE to begin flash boot immediately.
Flash boot interrupted.
0: i8255X @ PCI(bus:0 dev:13 irq:11)
1: i8255X @ PCI(bus:0 dev:14 irq:10)

Using 1: i82558 @ PCI(bus:0 dev:14 irq:10), MAC: 0090.2722.f0b1
Use ? for help.
monitor> addr 192.168.1.1
address 192.168.1.1
monitor> serv 192.168.1.2
server 192.168.1.2
monitor> file cdisk
file cdisk
monitor> ping 192.168.1.2
Sending 5, 100-byte 0x5b8d ICMP Echoes to 192.168.1.2, timeout is 4 seconds:
!!!!!
Success rate is 100 percent (5/5)
monitor> tftp
tftp pix512.bin@192.168.1.2.............................
Received 626688 bytes

PIX admin loader (3.0) #0: Mon Aug 7 10:43:02 PDT 1999
Flash=AT29C040A @ 0x300
Flash version 5.2.1, Install version 5.3.1

Installing to flash
…
```

TFTP Download Error Codes

During a TFTP download, if tracing is on, non-fatal errors appear in the midst of dots that display as the software downloads. The error code appears inside angle brackets. Table 2-2 lists the code values.

For example, random bad blocks appear as follows:

```
....<11>..<11>.<11>......<11>...
```

Also, tracing will show “A” and “T” for ARP and timeouts, respectively. Receipt of non-IP packets causes the protocol number to display inside parentheses.

Table 2-2 lists the TFTP error codes.

Table 2-2 Error Code Numeric Values

Error Code	Description
-1	Timeout between the PIX Firewall and TFTP server.
2	The packet length as received from the Ethernet device was not big enough to be a valid TFTP packet.
3	The received packet was not from the server specified in the **server** command.
4	The IP header length was not big enough to be a valid TFTP packet.
5	The IP protocol type on the received packet was not UDP, which is the underlying protocol used by TFTP.
6	The received IP packet's destination address did not match the address specified by the **address** command.
7	The UDP ports on either side of the connection did not match the expected values. This means either the local port was not the previously selected port, or the foreign port was not the TFTP port, or both.
8	The UDP checksum calculation on the packet failed.
9	An unexpected TFTP code occurred.
10	A TFTP transfer error occurred.
-10	The image file name you specified cannot be found. Check the spelling of the filename and that permissions permit the TFTP server to access the file. In UNIX, the file needs to be world readable.
11	A TFTP packet was received out of sequence.

Error codes 9 and 10 cause the download to stop.

Download the Latest Software from the Web

You can obtain PIX Firewall software by downloading it from Cisco's online web or FTP site. If you are using FTP, refer to the section "Download the Latest Software with FTP."

Before downloading software, you need to have a CCO username and password. If you do not have these, register now at the following site:

http://www.cisco.com/register/

Follow these steps to install the latest PIX Firewall software:

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 Access CCO at http://www.cisco.com and log in. Then access the PIX Firewall software downloads at the following site:

http://www.cisco.com/cgi-bin/tablebuild.pl/pix

Step 5 Obtain the software you need. If you have a PIX Firewall unit with a diskette drive, you must obtain the Boothelper binary image file bh512.bin so you can store a PIX Firewall image on a diskette. If you have a PIX 515, you can skip the discussion of the Boothelper diskette.

Download the Latest Software with FTP

Before using FTP, you need to have a CCO username and password. If you do not have these, register now at the following site:

http://www.cisco.com/register/

Once you have registered, 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.

Follow these steps 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 top level software 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.

Step 5 If you have not done so already, you can also download a TFTP server for use with Windows by using the **cd ..** command to return to the **internet** directory. Then use the **cd tftp** command to access the TFTP software directory. Use the **get** command to copy the TFTP executable file to your directory.

Step 6 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 the following site:

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

Step 7 When you are done, enter **quit** to exit.

Obtain the Bootheiper Binary Image

If your PIX Firewall unit has a diskette drive, you need to obtain the Boothelper binary image file bh521.bin and create a diskette.

This section contains the following topics:

- Get the Boothelper Binary Image
- Preparing a Boothelper Diskette With UNIX, Solaris, or LINUX
- Preparing a Boothelper Diskette on a Windows System

Get the Boothelper Binary Image

Use the following steps to download the Boothelper binary image:

Step 1 Log in to CCO and continue to the PIX Firewall software directory, as described in the previous section, "Download the Latest Software from the Web" or "Download the Latest Software with FTP."

Step 2 Download the bh521.bin Boothelper image from CCO and prepare a diskette as described in the sections that follow.

Note The Boothelper installation only supports PIX Firewall version 5.1, 5.2, 5.3, and later. After Boothelper downloads the PIX Firewall image via TFTP, it verifies the checksum of the image. If it is not version 5.1 or later, it displays the message "Checksum verification on flash image failed" and reboots the PIX Firewall.

Step 3 Download the PIX Firewall software binary image file pix521.bin from CCO and store this file in a directory accessible by your TFTP server.

Preparing a Boothelper Diskette With UNIX, Solaris, or LINUX

Follow these steps to prepare a Boothelper diskette:

Step 1 To prepare a UNIX, Solaris, or LINUX TFTP server to provide an image to the PIX Firewall, edit the inetd.conf file to remove the # (comment character) from the start of the "tftp" statement.

Step 2 Use the **ps aux | grep inetd** command string to determine the process ID of the current inetd process.

Step 3 Use the **kill -HUP** *process_id* command to kill the process. The process will restart automatically.

Step 4 Use the **dd** command to create the Boothelper diskette for the PIX Firewall unit. For example, if the diskette device name is rd0, use the following command:

```
dd bs=18b if=./bh510.bin of=/dev/rd0
```

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 rd0 on some UNIX systems.

Step 5 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.

Preparing a Boothelper Diskette on a Windows System

Follow these steps to create the Boothelper diskette from a Windows system:

Step 1 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.

Step 2 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. A sample **rawrite** session follows:

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

Enter source file name: bh512.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>
```

Ensure that the binary filename is in the "8.3" character format (8 characters before the dot; 3 characters after the dot).

Step 3 When you are done, 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.

Use Boothelper to Download an Image

Follow these steps to use the Boothelper diskette to download an image from a TFTP server:

Step 1 Download a PIX Firewall image from CCO (Cisco Connection Online) and store it on the host running the TFTP server.

Step 2 Start the TFTP server on the remote host and point the TFTP server to the directory containing the PIX Firewall image. On the Cisco TFTP Server, access the **View**>**Options** menu and enter the name of the directory containing the image in the **TFTP server root directory** field.

Step 3 Connect a console to the PIX Firewall and ensure that it is ready.

Step 4 Put the Boothelper diskette you prepared in the PIX Firewall and reboot it. When the PIX Firewall starts, the pixboothelper> prompt appears.

Step 5 You can now enter commands to download the binary image from the TFTP server. In most cases, you need only specify the **address**, **server**, and **file** commands, and then enter the **tftp** command to start the download. The commands are as follows:

a. If needed, use a question mark (**?**) or enter the **help** command to list the available commands.

b. Use the **address** command to specify the IP address of the network interface on which the TFTP server resides.

c. Use the **server** command to specify the IP address of the host running the TFTP server.

d. Use the **file** command to specify the filename of the PIX Firewall image.

e. If needed, use the **gateway** command to specify the IP address of a router gateway through which the server is accessible.

f. If needed, use the **ping** command to verify accessibility. If this command fails, fix access to the server before continuing. You can use the **interface** command to specify which interface the ping traffic should use. The Boothelper defaults to the interface 1 (one).

g. Use the **tftp** command to start the download.

Step 6 After the image downloads, you are prompted to install the new image. Enter **y**.

Step 7 When you are prompted, enter your activation key.

Step 8 After you enter your activation key, PIX Firewall prompts you to remove the Boothelper diskette. You have 30 seconds to remove the diskette. During this time you have three options:

a. Remove the diskette and reboot the unit with the reboot switch.

b. Use the **reload** command while the diskette is in the unit.

c. After the interval, the PIX Firewall will automatically boot from the Boothelper diskette.

After Boothelper downloads the PIX Firewall image via TFTP, it verifies the checksum of the image. If it is not version 5.1 or later, it displays the message “Checksum verification on flash image failed” and reboots the PIX Firewall.

Keep the Boothelper diskette available for future upgrades. You will need to repeat these steps whenever you download an image to your PIX Firewall unit. Alternatively, you can use the **copy tftp flash** command to download an image directly from the PIX Firewall command line.

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 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.

Follow these steps 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 At the PIX Firewall, access configuration mode by entering the **configure terminal** command.

Step 3 Also at the PIX Firewall, 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 From the router, set the default route to the inside interface of the PIX Firewall with the following Cisco IOS software command:

```
ip route 0.0.0.0 0.0.0.0 pix_inside_interface_ip_address
```

Step 6 While still at the router, enter the **show ip route** command and make sure that the PIX Firewall interface is listed as the "gateway of last resort."

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

Step 8 From the router, 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 other 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.

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

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**.

Follow these steps 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

Follow these steps 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, at 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.

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.

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 5.3* 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.

This section includes the following topics:

- The nameif Command
- The ip address Command
- The interface Command

The nameif Command

The PIX Firewall default configuration supplies **nameif** commands for the inside, outside and perimeter interfaces. Use the **show nameif** command to view these commands. An example **nameif** command follows:

```
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 perimeter security50
```

Note With the 5.3 software release, it is no longer necessary to use ethernet1 as the inside network port and ethernet2 outside network port. Any port, whether fixed or a PCI expansion port, and any interface type, FDDI, Token Ring, Fast Ethernet, or Gigabit Ethernet, can be assigned to be the inside or outside network port.

An example **nameif** command follows:

```
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 the following command:

```
nameif ethernot2 permetter security50
```

Reenter the corrected command as follows:

```
nameif ethernet2 perimeter security50
```

The **nameif** commands that need to be entered, if any, are determined by how many network interface cards are in your PIX Firewall.

Use the sections that follow depending on the number of interface cards:

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

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 default name is "inside" and the outside interface default name is "outside." Any perimeter interface default names are "intf*n*," such as "intf2" for the first perimeter interface, "intf3" for the second perimeter interface, and so on to the last interface. The numbers in the intf string corresponds to the interface card's position in the PIX Firewall. You can use the default names or give each interface a more meaningful name.

The format for the **nameif** command is as follows:

```
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 as follows:

```
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.

Note Always specify a network mask with the **ip address** command. If you let PIX Firewall assign a network mask based on the IP address, you may not be permitted to enter subsequent IP addresses if another interface's address is in the same range as the first address. For example, if you specify an inside interface address of 10.1.1.1 without specifying a network mask and then try to specify 10.1.2.2 for a perimeter interface mask, PIX Firewall displays the error message, "Sorry, not allowed to enter IP address on same network as interface *n*." To fix this problem, reenter the first command specifying the correct network mask.

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 follows:

```
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. If your PIX Firewall has gigabit Ethernet, FDDI, or Token Ring interfaces, refer to the **interface** command page in Chapter 5, "Command Reference," for configuration information.

Note 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 follows:

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*—For best performance, specify the speed of the interface; such as, **10baset** for 10 Mbps Ethernet half duplex communication, **10full** for 10 Mbps Ethernet full duplex communication, **100basetx** for 100 Mbps Ethernet half duplex communication, **100full** for 100 Mbps Ethernet full duplex communication, **1000sxfull** for 1000 Mbps Gigabit Ethernet full duplex operation, or **1000basesx** for 1000 Mbps Gigabit Ethernet half duplex operation. For Token Ring interfaces, auto-sensing is not supported; use either **4mbps** or **16mbps** depending on the speed of the interface. An interface speed is not specified for FDDI interfaces. Cisco recommends that you do not use the **auto** option.
- **shutdown**—Disables use of the interface. When you first install PIX Firewall, 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 10baset
  interface ethernet1 10baset
  interface ethernet2 10baset
  ```

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

Examples of the **interface** command are as follows:

```
interface ethernet0 10baset
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 **access-list** command described in the section "Step 13—Add Inbound 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.

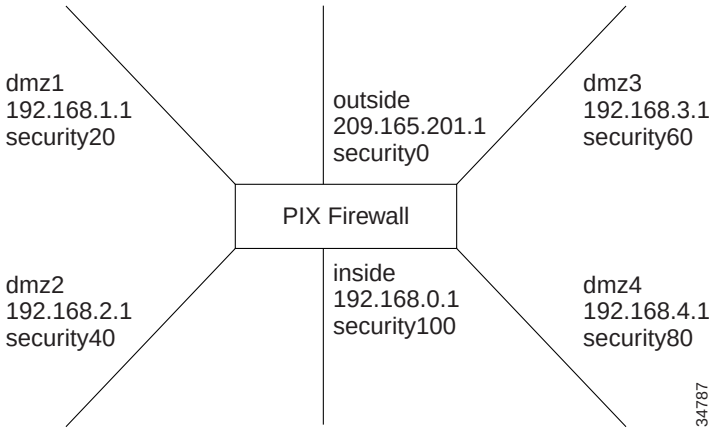
As you enter each command and debug it, you have to work with how your network addressing affects server access, creating global pools, authentication, routing, and starting connections. If you need to disable NAT, use the **nat 0** command. Refer to the **nat** command page, described in Chapter 5, "Command Reference," for how to disable NAT.

Follow these steps 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:

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

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

e. 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 the following:

```
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 65,535 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) 1 209.165.201.5 netmask 255.255.255.224
global (outside) 1 209.165.201.10-209.165.201.20 netmask 255.255.255.224
```

The first **global** command statement specifies a single IP address, which the PIX Firewall interprets as a PAT. You can specify PAT using the IP address at the interface using the **interface** keyword.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 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,dmz2, dmz3, and dmz4 start connections on the dmz1 interface.

The **global** command statement for dmz2 lets users on the inside, dmz3, and dmz4 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.

You can track usage among different subnets by mapping different internal subnets to different PAT addresses.

For example:

```
nat (inside) 1 10.1.0.0 255.255.0.0
nat (inside) 2 10.1.1.1 255.255.0.0
global (outside) 1 192.168.1.1
global (outside) 2 209.165.200.225
```

In this example, hosts on the internal network 10.1.0.0/16 are mapped to global address 192.168.1.1, and hosts on the internal network 10.1.1.1/16 are mapped to to global address 209.165.200.225 in global configuration mode.

Another way to measure traffic is to back up your PAT address.

For example:

```
nat (inside) 1 10.1.0.0 255.255.0.0
global (outside) 1 209.165.200.225
global (outside) 1 192.168.1.1
```

In this example, two port addresses are configured for setting up PAT on hosts from the internal network 10.1.0.0/16 in global configuration mode.

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 209.165.201.2, you would use this command:

```
route outside 0 0 209.165.201.2 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 209.165.201.2 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, in this case, on the outside interface.

You can only have one default route for the PIX Firewall.

Configuring the PIX Firewall to Work with Network Routing

Once a default route is set and the routers on the network have their default addresses set, as described in the section, "Step 3—Configure Network Routing," you need to configure the PIX Firewall with global addresses to direct traffic to the external networks and local addresses to direct traffic to the internal networks. For example, if you had a scenario where a PIX Firewall is on an intranet and has an inside network of 10.3.1.0 and an outside network of 10.42.1.0. The users you want to access the 10.3.1.0 network are on 192.168.1.0 network of the intranet. To permit these users access to the 10.3.1.0 network, you would use the following **static** command statement:

```
static (inside,outside) 10.42.1.0 10.3.1.0
```

Because the PIX Firewall is not a router, you need to configure static **route** command statements, such as the following static **route** statement, to direct traffic into the local network:

route inside 10.3.1.0 255.255.255.0 *next_hop_router_ip_address metric*

To route traffic to the outside network, you would use the following **route** command statement for the network of those who you want to access the network:

route outside 192.168.1.0 255.255.255.0 *next_hop_router_ip_address metric*

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.

Step 8—Permit Ping Access

You can use the **access-list** command to let you ping from a host on an interface through the PIX Firewall to a host on another interface. This lets you test that the host is reachable through the PIX Firewall.

The ping program sends an ICMP echo request message to the IP address and then expects to receive an ICMP echo reply. The ping program also measures how long it takes to receive the reply, which you can use to get a relative sense of how far away the host is.

Cisco recommends that you only permit pinging during troubleshooting and thereafter, disable pinging so the PIX Firewall unit is not visible on the network. Refer to “Disabling Interface Pinging” for more information.

This section includes the following topics:

- Configuring Ping Access
- Disabling Interface Pinging

Configuring Ping Access

Use an **access-list** command to permit ICMP access as follows:

```
access-list acl_out permit icmp any any
```

The “acl_out” is an **access-list** command ID and can be any name or a number you specify. Use the **show access-list** command to view this command in the configuration.

You then need to specify an **access-group** command for each interface through which you want the ICMP packets to pass. Use the **show access-group** command to view this command in the configuration.

To ping from one interface to another, bind the **access-list** and **access-group** command statements to the lower security interface, which lets the ICMP echo reply to return to the sending host.

For example, use the following command statement to ping from the inside interface to the outside interface:

```
access-group acl_out in interface outside
```

PIX Firewall only lets you bind one group of **access-list** command statements to an interface. Because you will need to add other **access-list** command statements to handle inbound server access (described in "Step 13—Add Inbound Server Access"), create unique groups for each interface. When you add additional **access-list** command statements use the same group ID as the **access-group** command statement for the respective interface.

For example, the following command statements let you ping from the inside interface to each of the other lower security interfaces:

```
access-list acl_dmz1 permit icmp any any
access-group acl_dmz1 in interface dmz1

access-list acl_dmz2 permit icmp any any
access-group acl_dmz2 in interface dmz2

access-list acl_dmz3 permit icmp any any
access-group acl_dmz3 in interface dmz3

access-list acl_dmz4 permit icmp any any
access-group acl_dmz4 in interface dmz4
```

Note We recommend you only open ICMP access on the interfaces you want to test. Having open ICMP access increases PIX Firewall operation overhead and can let attackers probe your network.

When you are done testing the interfaces, you can remove the ICMP **access-list** command statements from the configuration as follows:

```
no access-list acl_in permit icmp any any
no access-list acl_out permit icmp any any
no access-list acl_dmz1 permit icmp any any
no access-list acl_dmz2 permit icmp any any
no access-list acl_dmz3 permit icmp any any
no access-list acl_dmz4 permit icmp any any
```

You can also remove the **access-group** command statements, but be sure not to remove those associated with other **access-list** command statements.

An alternative to the **access-list** command is the **conduit** command, which is a legacy PIX Firewall command. The **access-list** command supersedes the **conduit** command.

You can use the following **conduit** command to open all interfaces for ping use:

```
conduit permit icmp any any
```

Disable this command after testing with the following command statement:

```
no conduit permit icmp any any
```

You can view **conduit** commands in your configuration with the **show conduit** command statement.

Note We recommend you only use the **access-list** and **access-group** commands to maintain future compatibility with PIX Firewall software and compatibility with Cisco IOS software.

Disabling Interface Pinging

With pinging disabled, the PIX Firewall cannot be detected on the network. The new **icmp** command implements this feature. This feature is also referred to as configurable proxy pinging. To disable pinging, first configure an **access-list** command statement that permits or denies ICMP traffic that terminates at the PIX Firewall unit, and then add the appropriate **icmp** command statement to your configuration.

The **icmp** commands are:

icmp permit | **deny** [**host**] *src_addr* [*src_mask*] [*type*] *int_name*

no icmp permit | **deny** [**host**] *src_addr* [*src_mask*] [*type*] *int_name*

clear icmp

show icmp

where:

permit | **deny**	Permit or deny the ability to ping a PIX Firewall interface.
src_addr	Address that is either permitted or denied ability to ping an interface. Use **host** *src_addr* to specify a single host.
src_mask	Network mask. Specify if a network address is specified.
type	ICMP message type.
int_name	Interface name that can be pinged.

If the first matched entry is a permit entry, the ICMP packet continues to be processed. If the first matched entry is a deny entry or an entry is not matched, PIX Firewall discards the ICMP packet and generates the %PIX-3-313001 syslog message. An exception is when an ICMP **access-list** command statement is not configured; then, permit is assumed.

Cisco recommends that you grant permission for ICMP unreachable message type (type 3). Denying ICMP unreachable messages, disables ICMP Path MTU discovery, which can halt IPSec and PPTP traffic. See RFC 1195 and RFC 1435 for details about Path MTU Discovery.

The syslog message is as follows:

%PIX-3-313001: Denied ICMP type=*type*, code=*code* from *source_address* on interface *interface_number*

If this message appears, contact the peer's administrator.

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 reboot the PIX Firewall. When you reboot, all traffic through the PIX Firewall stops. Once the PIX Firewall unit is again available, connections can restart. 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) 1 209.165.201.4 netmask 255.255.255.224
```

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) 1 209.165.201.3 netmask 255.255.255.224
global (outside) 1 209.165.201.10-209.165.201.20 netmask 255.255.255.224
global (dmz1) 1 192.168.1.20-192.168.1.254 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 with IP address 209.165.201.3. The second **global** command statement creates a pool of IP addresses in the range of 209.165.201.10 to 209.165.201.20 on the outside interface.

The third **global** command statement creates a pool of IP addresses on the dmz1 interface in the range of 192.168.1.20 to 192.168.1.254.

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.

Step 6 If your ISP (Internet service provider) has only provided a few registered addresses, always include a PAT address. This expands your pool of addresses, if needed.

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 255.255.255.240 subnet mask, the pool of global addresses could not contain addresses 209.165.201.10 to 209.165.201.20 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 255.255.255.240 mask, specifying a global pool of 209.165.201.16 to 209.165.201.31 would not work because 209.165.201.16 is a network address and 209.165.201.31 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 the registered address 209.165.201.1 with a 255.255.255.224 subnet mask, the hosts on the outside interface, the outside router, the global pool, and any addresses set aside for **static** command statements (explained in “Step 13—Add Inbound Server Access”) must all have addresses in this subnet in the range of 209.165.201.2 through 209.165.201.30.

b. If you are using subnetting, put the subnet value in the command statements that let you specify a mask. For example, if you are using a .224 subnet mask, the **ip address** command would appear as follows:

```
ip address outside 209.165.201.1 255.255.255.224
```

The **global** command would appear as:

```
global (outside) 1 209.165.201.10-209.165.201.30 netmask 255.255.255.224
```

Step 8 Use the **show nat** command to view **nat** command statements in your configuration. 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.

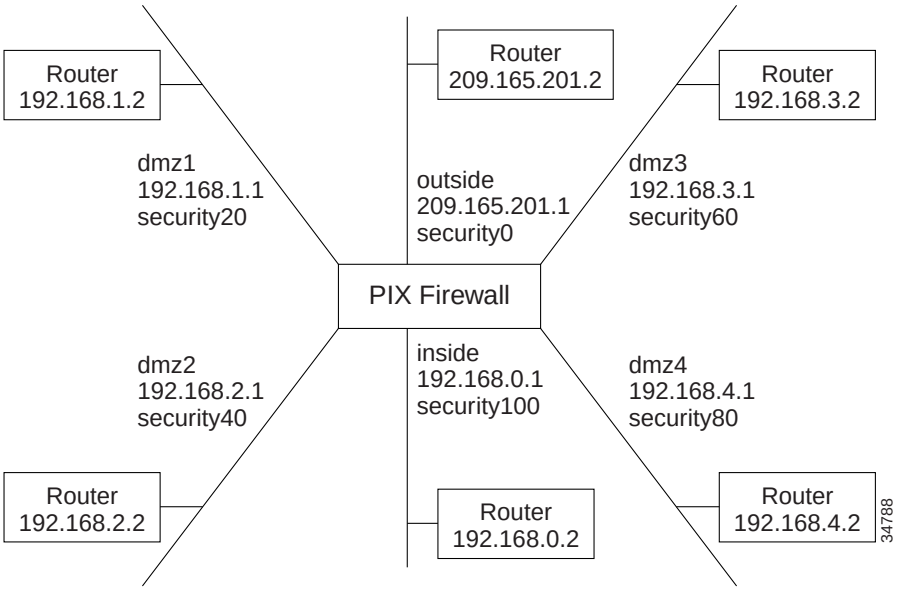
Step 11—Test Network Connectivity

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 unit's interface. In version 5.3, the PIX Firewall **ping** command has been improved so do not need to specify the interface name if the host's IP address is on the same subnet as a PIX Firewall interface. For the example, you would use these **ping** commands from the PIX Firewall command line to ping hosts or routers:

```
ping 192.168.0.2
ping 192.168.1.2
ping 192.168.2.2
ping 192.168.3.2
ping 192.168.4.2
ping 209.165.201.2
```

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

- Ping the PIX Firewall's outside interface with **`ping 209.165.201.1`**
- 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 reply (len 32 id 1 seq 256) 209.165.201.1 > 209.165.201.2
ICMP echo request (len 32 id 1 seq 512) 209.165.201.2 > 209.165.201.1
```

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 0 209.165.201.2 1
```

b. Use the **show access-list** command to ensure that you have **access-list** command statements in your configuration to permit ICMP. Add these commands if they are not present. Refer to "Step 8—Permit Ping Access" for more information.

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.

Step 4 **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:

```
Outbound ICMP echo request (len 32 id 1 seq 512) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 256) 209.165.201.1 > 209.165.201.2
```

Step 5 **Add static and access-list command statements and test again**—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 **access-list** command statements as described in the section "Step 13—Add Inbound 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 IPSec configured, you can use Telnet to remotely administer the console of a PIX Firewall from the outside interface. Refer to "Securing a Telnet Connection on the Outside Interface" for more information.

This section contains the following sections:

- Configuring Telnet Console Access
- Securing a Telnet Connection on the Outside Interface
- Trace Channel Feature

Configuring Telnet Console Access

Follow these steps 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. Refer to "Securing a Telnet Connection on the Outside Interface" for more information. Use a command such as the following:

```
telnet 209.165.200.225 255.255.255.224 outside
```

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.

Follow these steps 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 from a successful ping appear as follows:

```
Outbound ICMP echo request (len 32 id 1 seq 512) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 256) 209.165.201.1 > 209.165.201.23
```

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.

Securing a Telnet Connection on the Outside Interface

This section tells you how to secure your PIX Firewall console Telnet connection to the outside interface of the PIX Firewall. If you are using the Cisco Secure Policy Manager, version 2.0 or later, this section also applies to you. It is assumed you are using the Cisco Secure VPN Client, version 1.1 or the Cisco VPN 3000 Client version 2.5, to secure your Telnet connection.

See the **telnet** command page within Chapter 5, "Command Reference," for more information about this command.

For IPSec information, refer to the *IPSec User Guide for the Cisco Secure PIX Firewall Version 5.3*.

Note You will need to have two security policies set up on your VPN Client. One security policy is used to secure your Telnet connection and another to secure your connection to the inside network.

To encrypt your Telnet connection to the PIX Firewall's outside interface, perform the following steps as part of your PIX Firewall configuration. In the following examples, the IP address of the PIX Firewall's outside interface is 168.20.1.5, and the VPN Client's IP address stemming from the virtual pool of addresses is 10.1.2.0.

Step 1 Create an **access-list** command statement to define the traffic to protect from the PIX Firewall to the VPN Client using a destination address from the virtual local pool of addresses:

```
access-list 80 permit ip host 168.20.1.5 10.1.2.0 255.255.255.0
```

Step 2 Specify which host can access the PIX Firewall console with Telnet. Specify the VPN Client's address from the local pool and the outside interface:

```
telnet 10.1.2.0 255.255.255.0 outside
```

Step 3 Within the VPN Client, create a security policy that specifies the Remote Party Identity IP address and gateway IP address as the same IP address—the IP address of the PIX Firewall's outside interface. In this example, the IP address of the PIX Firewall's outside is 168.20.1.5.

Configure the rest of the security policy on the VPN Client to match the PIX Firewall's security policy.

Trace Channel Feature

The **debug packet** command sends its output to the Trace Channel. All other **debug** commands do not. Use of Trace Channel changes the way you can view output on your screen during a PIX Firewall console or Telnet session.

If a **debug** command does not use Trace Channel, each session operates independently, which means any commands started in the session only appear in the session. By default, a session not using Trace Channel has output disabled by default.

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** 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 becomes the Trace Channel. The next Telnet console session that accesses the console then becomes the Trace Channel.

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** 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 output is not appearing, use the **who** command to see if a Telnet console session is running.

Step 13—Add Inbound Server Access

By default, the PIX Firewall prevents all outside connections from accessing "inside" hosts or servers. Use the **static**, **access-list**, and **access-group** command statements to permit access.

Note If you are using the **nat 0** (disable NAT) command, refer to the **static** command page in Chapter 5, "Command Reference," for configuration information.

To add server access, use these commands:

- **static**—Provides an IP address that users on a less secure interface can use to access a server on a more secure interface. Use this rule for creating **static** command statements:

  static (*high_interface***,***low_interface***)** *low_address high_address* **netmask** *netmask*

  In the parentheses, specify the two interfaces to be accessed. Always specify the most secure (*high_interface*) first followed by the lesser secure interface (*low_interface*). For example, to let users on the outside interface access a web server on the dmz3 interface, you would code **(dmz3,outside)** because the dmz3 interface is more secure than the outside interface.

  You can view the security of each interface with the **show nameif** command. The higher the security number, the more secure the interface. The outside interface is always the least secure (shown as security0) and the inside interface is always the most secure (security100).

  The next two parameters are for IP addresses. Specify the global address you want users to access for *low_address* and the address of the server as *high_address*. The global address you specify is one you make up. The only requirements for the IP address you specify are that it be on the same subnet as the interface's address and that it not be used in any other command statement or be for another device in your network. The *high_address* address is the actual IP address of the server.

  Use a netmask of 255.255.255.255 when low_address is a host even if subnetting is in place, or specify the correct mask if low_address is a network address.

  For example, using the addresses in Figure 2-3, specify a global address on the outside interface as 209.165.201.3 that you want outside users to access when they send requests to the dmz3 web server at 192.168.3.3. The **static** command statement would be as follows:

  ```
  static (dmz3,outside) 209.165.201.3 192.168.3.3 netmask 255.255.255.255
  ```

- **access-list**—Defines permissions for how users can access the global address; such as, the IP addresses of users who can access the global address and what port they are allowed to access. The format of the **access-list** command is as follows:

  access-list *ID action protocol source_address port destination_address port*

  The parameters are as follows:

  – *ID*—A name or number you create to identify a group of **access-list** command statements; for example, "acl_out," which identifies that the permissions apply to access from the outside interface.

  – *action*—Either **permit** or **deny** depending on whether you want to permit or deny access to the server. By default, all inbound access is denied, so you will need to permit access to a specific protocol or port.

  – *protocol*—Values are listed in "Protocols" in Chapter 1, "Introduction." For most servers, such as for the Web or email, use **tcp**.

– *source_address*—A host or network address for those systems on the lower security level interface that need to access the *destination_address*, which is the *low_address* in the **static** command statement. Use **any** for the *source_address* to let any host access the *destination_address*. If you specify a single host, precede the address with **host**; for example **`host 192.168.1.2`**. If you specify a network address, also specify a network mask; for example, **`192.168.1.0 255.255.255.0`**.

  If you are familiar with Cisco IOS software, note that PIX Firewall uses a subnet mask, whereas Cisco IOS software uses a wildcard mask. (In Cisco IOS software, the mask in this example would be specified with the **`0.0.0.255`** value.)

– *destination_address*—A host or network global address that you specified with the **static** command statement. For a host address, precede the address with **host**; for networks, specify the network address and the appropriate network mask.

– *port*—Values are listed in "Ports" in Chapter 1, "Introduction." For a web server, use **www** for port 80. For an email server, use **smtp** for port 25. The port is preceded with the **eq** (equals) parameter.

To let any users on the outside interface access the dmz3 web server, the **access-list** command statement with the previously shown **static** command statement is as follows:

```
static (dmz3,outside) 209.165.201.3 192.168.3.3 netmask 255.255.255.255
access-list acl_out permit tcp any host 209.165.201.3 eq www
```

- **access-group**—Binds the **access-list** command statement to a PIX Firewall interface. The format of the **access-group** command is as follows:

  access-group *ID* **in interface** *low_interface*

  The *ID* is the same name you specified in the **access-list** command statement. The *low_interface* is the lower security interface you specified in the **static** command statement; that is, the interface on which users will access the global address. Only specify one **access-group** command for each interface.To let users on the outside interface access the dmz3 web server, the **access-group** command with the other two command statements is as follows:

```
static (dmz3,outside) 209.165.201.3 192.168.3.3 netmask 255.255.255.255
access-list acl_out permit tcp any host 209.165.201.3 eq www
access-group acl_out in interface outside
```

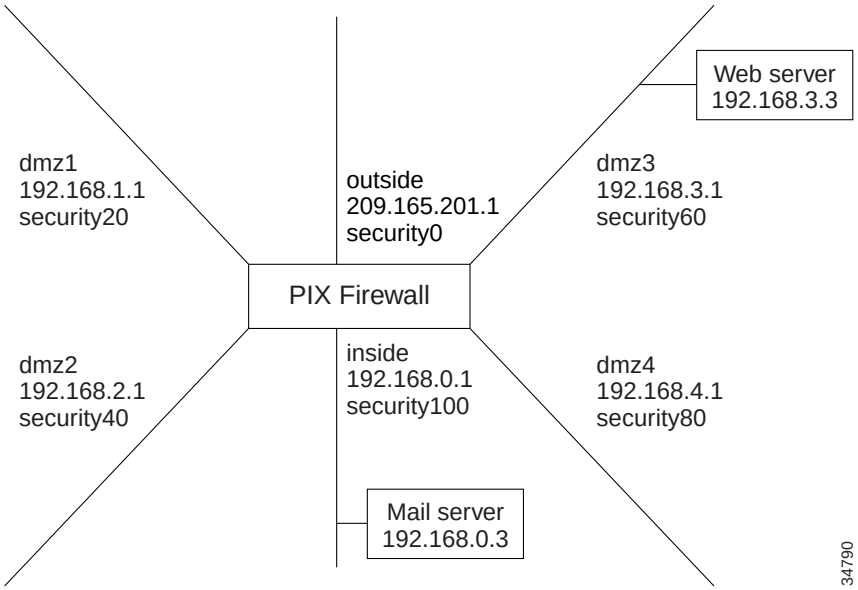
Configuring for a Server

Follow these steps 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) 209.165.201.4 192.168.0.3 netmask 255.255.255.255
access-list acl_out permit tcp any host 209.165.201.4 eq smtp
access-group acl_out in interface outside
```

These commands create a global address of 209.165.201.4 that PIX Firewall maps to the 192.168.0.3 mail server on the dmz2 interface. The **access-list** command statement permits any outside users to access the mail server at the SMTP port (25). The **access-group** command statement binds the mail server permission to the outside interface.

You will need to inform your DNS administrator to create an MX record for the global address (such as 209.165.201.4) so that mail is directed to the correct address.

Two **access-list** 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 **access-list** command statement for TCP and one for UDP. For DNS, if you are only receiving zone updates, you only need a single **access-list** command statement for TCP.

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

```
static (dmz2,outside) 209.165.201.5 192.168.1.5 netmask 255.255.255.255
access-list acl_out permit tcp any host 209.165.201.5 eq 1723
access-list acl_out permit gre any host 209.165.201.5
access-group acl_out in interface outside
```

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 209.165.201.5. The first **access-list** command statement opens access for the PPTP protocol and gives access to any outside users. The second **access-list** permits access to GRE. If PPTP was not involved and GRE was, you could omit the first **access-list** command statement.

Step 4 Add the remaining **static** and **access-list** command statements:

a. 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
access-list acl_dmz1 permit tcp any host 192.168.1.4 eq smtp
access-group acl_dmz1 in interface dmz1
```

b. To let dmz2 users access the mail server, enter the following:

```
static (inside,dmz2) 192.168.2.4 192.168.0.3 netmask 255.255.255.255
access-list acl_dmz2 permit tcp any host 192.168.2.4 eq smtp
access-group acl_dmz2 in interface dmz2
```

c. To let dmz3 users access the mail server, enter the following:

```
static (inside,dmz3) 192.168.3.4 192.168.0.3 netmask 255.255.255.255
access-list acl_dmz3 permit tcp any host 192.168.3.4 eq smtp
access-group acl_dmz3 in interface dmz3
```

d. To let dmz4 users access the mail server, enter the following:

```
static (inside,dmz4) 192.168.4.4 192.168.0.3 netmask 255.255.255.255
access-list acl_dmz4 permit tcp any host 192.168.4.4 eq smtp
access-group acl_dmz4 in interface dmz4
```

These command statements create a global address on each interface to map to the inside mail server and then create an access list 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 192.168.0.3, 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.

Follow these steps 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.

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

```
static (dmz3,outside) 209.165.201.3 192.168.3.3 netmask 255.255.255.255
access-list acl_out permit tcp any host 209.165.201.3 eq www
access-group acl_out in interface outside
```

b. To let dmz1 users access the web server, enter the following commands:

```
static (dmz3,dmz1) 192.168.1.3 192.168.3.3 netmask 255.255.255.255
access-list acl_dmz1 permit tcp any host 192.168.1.3 eq www
access-group acl_dmz1 in interface dmz1
```

c. To let dmz2 users access the web server, enter the following commands:

```
static (dmz3,dmz2) 192.168.2.3 192.168.3.3 netmask 255.255.255.255
access-list acl_dmz2 permit tcp any host 192.168.2.3 eq www
access-group acl_dmz2 in interface dmz2
```

d. 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
```

e. 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 **access-list** 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 **access-list** 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. Users on the inside and dmz4 interfaces can access the web server at address 192.168.3.3, as would users on the same interface, dmz3. Users on dmz1 would access it at 192.168.1.3; users on dmz2 would access it at 192.168.2.3, and users on the outside would access it at 209.165.201.3.

Step 14—Add Outbound Access Lists

An outbound access list lets you restrict users from starting outbound connections or lets you restrict users from accessing an address. The "outbound" term means a connection started from a higher security level interface for access to a lower security level interface. The **show nameif** command lists the security levels for each interface.

For example, you could restrict some users from accessing web sites, permit others access, or restrict one or more users from accessing a specific web site. By default, outbound connections are permitted. Outbound access is restricted with the **access-list** command, which lets you selectively deny or permit access as required. Then use the **access-group** command to bind the **access-list** command statements to an interface.

The format for the **access-list** command in this context is as follows:

access-list *ID action protocol source_address src_port destination_address dest_port*

The parameters are as follows:

- *ID*—A name or number you create to identify a group of **access-list** command statements; for example, "acl_out," which identifies that the permissions apply to access from the outside interface.
- *action*—Either **permit** or **deny** depending on whether you want to permit or deny access to the server. By default, outbound access is permitted, so you will need to deny access to a specific protocol, address, or destination port.
- *protocol*—Values are listed in "Protocols" in Chapter 1, "Introduction." For most servers, such as for the Web or email, use **tcp**.
- *source_address*—A host or network address of those systems that need to access the *destination_address*. For outbound connections, the *source_address* is on a higher security level interface and the *destination_address* is on a lower security level interface. Use **any** instead of the *source_address* to permit or deny any host access to *destination_address*. If you specify a single host, precede the address with **host**; for example **`host 10.1.1.1`**. If you specify a network address, also specify a network mask; for example, **`10.1.2.0 255.255.255.0`**.

  If you are familiar with Cisco IOS software, note that the network mask octets are specified in reverse order in the PIX Firewall **access-list** command. (In Cisco IOS software, the mask in this example would be specified with the **`0.0.0.255`** value.)
- *src_port*—Rarely used. For outbound access lists, specify the port you want to restrict access to after the destination address.
- *destination_address*—A host or network address to which you deny or permit access on a lower security level interface.
- *dest_port*—Values are listed in "Ports" in Chapter 1, "Introduction." For a web server, use **www** for port 80. For an email server, use **smtp** for port 25. The port is preceded with the **eq** (equals) parameter.

This section includes the following topics:

- Restricting Users from Starting Connections
- Restricting Users from Accessing a Specific Server
- Filtering Outbound Connections

Restricting Users from Starting Connections

One of the uses of restricting connections is to prevent users from starting outbound connections. Because connections from a higher security level interface to a lower security level interface are permitted by default, you need to use the **access-list** command to deny access.

The **access-list** command statements are bound by the **access-group** command statement to a particular interface, and the **access-group** command will only filter packets going into the interface.

Denying users from starting connections means that you deny access into the higher security level interface and out on lower security level interface. In other words, specify the source address as either a host or network address on the higher security level interface and specify the destination address on the interface that you want to restrict access to.

For example, to prevent users on the 192.168.1.0 network on the inside interface from starting connections on the outside interface and permit all others, specify the 192.168.1.0 network address as the source address and the network connected to the outside interface as the destination address. In the example that follows, the network on the outside interface is 209.165.201.0. The **access-list** and **access-group** command statements are as follows:

```
access-list acl_in deny tcp 192.168.1.0 255.255.255.224 209.165.201.0 255.255.255.224
access-list acl_in permit ip any any
access-group acl_in in interface inside
```

In the next example, dmz1 interface users are restricted from web browsing on other interfaces, but one host at 192.168.1.2 has web access. Put the port you want to restrict users from after the destination address.

The following example shows these commands:

```
access-list acl_dmz1 deny tcp any any eq www
access-list acl_dmz1 deny tcp host 192.168.1.2 any eq www
access-group acl_dmz1 in interface dmz1
```

The first **access-list** command statement disables web access. The second **access-list** command statement lets host 192.168.1.2 web browse. The **access-group** command statement binds the **access-list** command statement to the dmz1 interface.

Note Access lists work on a first-match basis, so for outbound access lists, you need to permit first and then deny after.

Restricting Users from Accessing a Specific Server

You can use the **access-list** command to specify a server that users cannot access. For example, if you want to restrict users on the inside interface from accessing a web site at address 209.165.201.29 on the outside interface that has objectionable material, use the following commands:

```
access-list acl_in deny tcp any host 209.165.201.29 eq www
access-group acl_in in interface inside
```

These commands let any users start connections, but not to 209.165.201.29. The **access-group** command specifies that the users are on the inside interface.

In the next example, users in the 192.168.2.0 network on the dmz2 interface are restricted from accessing any servers on the dmz1 interface, which is on the 192.168.1.0 network:

```
access-list acl_dmz2 deny tcp 192.168.2.0 255.255.255.0 192.168.1.0 255.255.255.0
access-group acl_dmz2 in interface dmz2
```

Filtering Outbound Connections

ActiveX objects and Java applets are security risks for outbound connections because they can contain code to attack hosts and servers. You can disable ActiveX objects and remove Java applets with the PIX Firewall **filter** command. In addition, you can use the **filter** command to work with a Websense server to remove URLs you deem inappropriate for use at your site.

This section includes the following topics:

- Filtering ActiveX Objects
- Filtering Java Applets
- Filtering URLs with Websense

Filtering ActiveX Objects

ActiveX controls, formerly known as OLE or OCX controls, are components you can insert in a web page or other application. These controls include custom forms, calendars, or any of the extensive third-party forms for gathering or displaying information. As a technology, ActiveX creates many potential problems for the network clients including causing workstations to fail, introducing network security problems, or being used to attack servers.

The PIX Firewall ActiveX feature blocks the HTML <object> commands by commenting them out within the HTML web page. This functionality has been added to the **filter** command with the **activex** option.

Note The <object> tag is also used for Java applets, image files, and multimedia objects, which will also be blocked by the new command.

If the <object> or </object> HTML tags split across network packets or if the code in the tags is longer than the number of bytes in the MTU, PIX Firewall cannot block the tag.

Filtering Java Applets

The **filter java** command filters out Java applets that return to the PIX Firewall from an outbound connection. The user still receives the HTML page, but the web page source for the applet is commented out so that the applet cannot execute. Use 0 for the *local_ip* or *foreign_ip* IP addresses to mean all hosts.

Note If Java applets are known to be in <object> tags, use the **filter activex** command to remove them.

Examples

To specify that all outbound connections have Java applet blocking, use the following command:

```
filter java 80 0 0 0 0
```

This command specifies that the Java applet blocking applies to Web traffic on port 80 from any local host and for connections to any foreign host.

Filtering URLs with Websense

The **filter url** command lets you prevent outbound users from accessing World Wide Web URLs that you designate using the Websense filtering application.

The **allow** option to the **filter** command determines how the PIX Firewall behaves in the event that the Websense server goes offline. If you use the **allow** option with the **filter** command and the Websense server goes offline, port 80 traffic passes through the PIX Firewall without filtering. Used without the **allow** option and with the server offline, PIX Firewall stops outbound port 80 (Web) traffic until the server is back online, or if another URL server is available, passes control to the next URL server.

Note With the **allow** option set, PIX Firewall now passes control to an alternate server if the Websense server goes offline.

This section contains the following topics:

- Filtering URLs
- Websense Filtering by Username and Group
- Websense Information

Filtering URLs

Follow these steps to filter URLs:

Step 1 Designate a Websense server with the **url-server** command.

Step 2 Enable filtering with the **filter** command.

Step 3 If needed, improve throughput with the **url-cache** command. However, this command does not update Websense logs, which may affect Websense accounting reports. Accumulate Websense run logs before using the **url-cache** command.

Step 4 Use the **show url-cache stats** and the **show perfmon** commands to view run information.

Examples

The following example filters all outbound HTTP connections except those from the 10.0.2.54 host:

```
url-server (perimeter) host 10.0.1.1
filter url http 0 0 0 0
filter url except 10.0.2.54 255.255.255.255 0 0
```

Websense Filtering by Username and Group

The Websense Server (UFS) works with the PIX Firewall to deny users from access to web sites based on the company security policy.

Websense protocol version 4 enables group and username authentication between a host and a PIX Firewall. The PIX Firewall performs a username lookup, and then the Websense server handles URL filtering and username logging.

Websense protocol version 4 contains the following enhancements:

- URL filtering allows the PIX Firewall to check outgoing URL requests against the policy defined on the Websense server.
- Username logging tracks username, group, and domain name on the Websense server.
- Username lookup enables the PIX Firewall to use the user authentication table to map the host's IP address to the username.

Websense Information

Information on Websense is available at the following site:

http://www.websense.com/products/websense/

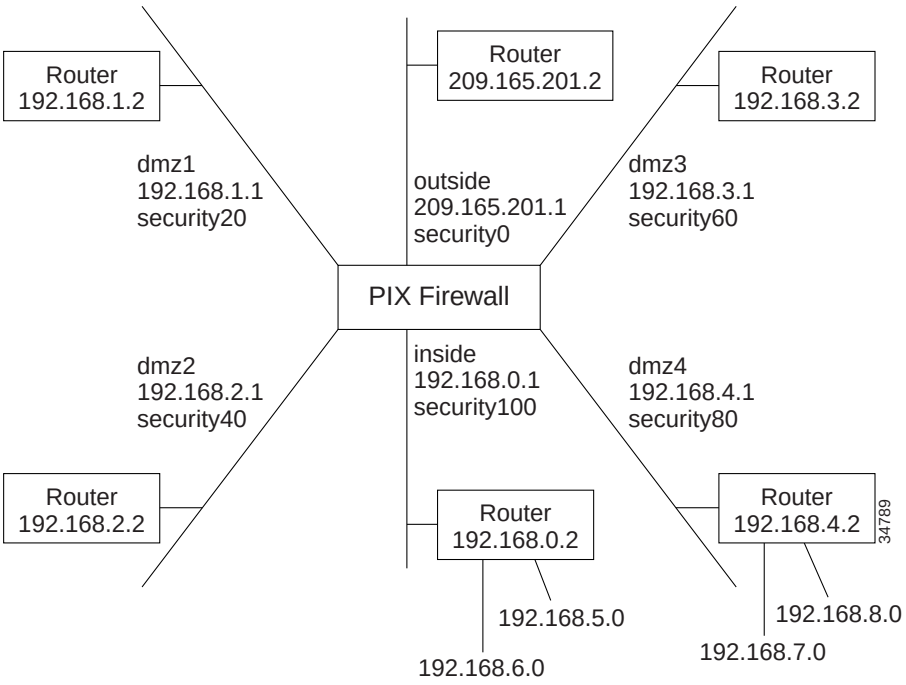
Step 15—Add Static Routes

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.

Follow these steps 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 Only one default route is permitted:

```
route outside 0 0 209.165.201.2 1
```

This command statement sends any packets destined for the default route, IP address 0.0.0.0 (abbreviated as **0**, and **0** for the netmask), to the router 209.165.201.2. 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 192.168.4.2.

Step 16—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.

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.

This section includes the following topics:

- Syslog Levels
- Viewing Messages from the Console
- Viewing Messages from a Telnet Console Session
- Sending Messages to a Syslog Server
- PIX Firewall Syslog Server Use
- Configuring a UNIX System for Syslog
- FTP and URL Logging

In addition, refer to "IDS Syslog Messages" in Chapter 3, "Advanced Configurations," for information on how to view Cisco Secure Intrusion Detection System (IDS) signatures.

Syslog Levels

Common to all ways to view syslog messages is the level, or severity, of a syslog 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.

Table 2-3 lists syslog message levels.

Table 2-3 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

System log messages received at a syslog server begin with a percent sign (%) and are structured as follows:

`%PIX-level-message_number: message_text`

You can set the *level* with the logging command so you can view syslog messages on the PIX Firewall console, from a syslog server, or with SNMP.

Refer to *System Log Messages for the Cisco Secure PIX Firewall Version 5.3* for information on each syslog *message_number* and the *message_text*. You can view this document online at the following site:

http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_v52/syslog/index.htm

Viewing Messages from the Console

Follow these steps 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 *System Log Messages for the Cisco Secure PIX Firewall Version 5.3* 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 **access-list** is in your configuration by using the **show access-list** command and checking for command statements similar to the following:

```
access-list acl_grp permit icmp any any
access-group acl_grp in interface lower_security_interface
```

If these commands are not present, then add them. The *acl_grp* parameter is a name you specify to associate the **access-list** command statement to the **access-group** command statement. The *lower_security_interface* is the lower security interface through which you are pinging a host on that interface. The **access-list** command statement permits echo replies to return to the host from which you initiated the ping.

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 follows:

```
logging buffered alerts
```

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

Viewing Messages from a Telnet Console Session

Follow these steps 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 192.168.1.1
```

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 a syslog server such as those in UNIX or other operating systems. If you have a Windows NT system available for use as a syslog server, you can use the PIX Firewall Syslog Server (PFSS). Refer to "PIX Firewall Syslog Server Use" for more information.

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.

Follow these steps 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
```

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." 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. Because network devices share the eight facilities, the **logging facility** command lets you set the facility for receiving PIX Firewall syslog messages.

Note Cisco recommends that you do not specify the **logging facility** command or change the local4 selector unless this value conflicts with another device generating syslog messages to the syslog server.

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 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 5.3*. You can access PIX Firewall documentation online at:

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

PIX Firewall Syslog Server Use

The PIX Firewall Syslog Server (PFSS) lets you view syslog messages from a Windows NT system.

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 5.3*.

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.

This section contains the following topics:

- Configuring PFSS
- Changing PFSS Parameters at the Windows NT System
- Recovering from PFSS Disk-full

Configuring PFSS

Use the following steps to configure for PFSS use:

Step 1 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.

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 *System Log Messages for the Cisco Secure PIX Firewall Version 5.3*.

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.

If you are using IPSec digital certificates, set the clock to Greenwich Mean Time (GMT). PIX Firewall does not have a provision for setting timezones and using GMT lets digital certificates work correctly.

Changing PFSS Parameters at the Windows NT System

You can change PFSS (PIX Firewall Syslog Server) parameters at the Windows NT system by clicking **Start**>**Settings**>**Control Panel**>**Services**.

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.

Follow these steps 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 PFSS 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.)

Follow these steps 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.

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.

Follow these steps 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 in to 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.*level*:

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

Choose a selector from Table 2-4.

Table 2-4 syslog.conf Selector Levels

Syslog Level	Use this selector in syslog.conf
0 - Emergencies	**local4.emerg**
1 - Alerts	**local4.alert**
2 - Critical	**local4.crit**
3 - Errors	**local4.err**
4 - Warnings	**local4.warn**
5 - Notifications	**local4.notice**
6 - Information	**local4.info**
7 - Debugging	**local4.debug**

Refer to the UNIX **syslog**(3) command page for more information on possible selectors.

Note You can use a different selector than local4, but if you change it to one of the other possibilities of LOCAL0 - LOCAL7, you must change the **logging facility** *nn* command accordingly. You then must set the appropriate number in the /etc/syslog.conf file. Cisco recommends that you not change the local4 selector unless this value conflicts with another device that generates syslog messages.

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 -HUP 'cat /etc/syslog.pid'
```

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

FTP and URL Logging

You can log FTP commands and WWW URLs when syslog is enabled. FTP and URL messages are logged at syslog level 7. Usernames are provided in the log information. Both inbound and outbound FTP commands and URLs are sent to syslog.

This section includes the following topics:

- Logging FTP and URL Messages
- Sample URL Log
- Sample FTP Log

Logging FTP and URL Messages

Use the following steps to enable FTP and URL logging:

Step 1 Use the **show fixup** command to ensure that the following **fixup protocol** commands for FTP and HTTP are present in the configuration:

```
fixup protocol http 80
fixup protocol ftp 21
```

These commands are in the default configuration.

Step 2 Enable URL logging by setting the **logging** command to level 5; set FTP logging by setting the **logging** command to level 6. Table 2-5 lists the **logging** commands that set the logging level.

Table 2-5 FTP and URL Logging Commands

logging Command	Description	View the Log:
logging buffered *n*	Send syslog messages to an internal buffer. Use the **clear logging** command to clear the message buffer. New messages appear at the end of the buffer.	With the **show logging** command.
logging console *n*	Send syslog messages to the console. Cisco recommends that you do not use this command in production mode because its use degrades PIX Firewall performance.	At the PIX Firewall console as they occur.

Table 2-5 FTP and URL Logging Commands

logging Command	Description	View the Log:
logging history *n*	Set the SNMP message level for sending syslog traps.	With an SNMP management station.
logging trap *n*	Set logging level only for syslog messages being sent to a syslog server.	At the syslog server.

The sections that follow provide sample output displays for each logging type.

Sample URL Log

The following is an example of a URL logging syslog message:

```
%PIX-5-304001: user 192.168.69.71 Accessed URL 10.133.219.25 : www.example.com
```

Sample FTP Log

The following is an example of an FTP logging syslog message:

```
%PIX-6-303002: 192.168.69.71 Retrieved 10.133.219.25: 10.1.1.42
```

You can view these messages at the PIX Firewall console with the **show logging** command.

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.

This section includes the following topics:

- Configuring for AAA
- Configuring RADIUS Authorization

Configuring for AAA

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.

Follow these steps to provide user authentication and authorization:

Step 1 For inbound authentication, create the **static** and **access-list** command statements required to permit outside hosts to access servers on the inside network, as described in "Step 13—Add Inbound 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 authorization is provided with the **access-list** command statement as described in "Configuring RADIUS 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 tacacs+
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 TACACS+ 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 include any outbound 0 0 0 0 AuthOutbound
aaa authentication include 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. PIX Firewall checks the authorization request with the AAA server, which makes the decision about what services a user can access. Use one or both of the following commands to specify outbound and inbound authorization:

```
aaa authorization include any outbound 0 0 0 0
aaa authorization include any inbound 0 0 0 0
```

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

aaa authorization include | **exclude** *author_service*|[*protocol*/*port*[**-***port*]] **inbound** | **outbound** | *if_name local_ip local_mask foreign_ip foreign_mask*

where:

- *author_service*—The service that PIX Firewall listens to for AAA connections. Possible values are **any**, **http**, **ftp**, or **telnet**.
- *protocol*—The protocol to authorize access to. Possible values are **udp**, **tcp**, or **icmp**.
- *port*—A port value or range to authorize users access to.
- **inbound**, **outbound**, *if_name*—Specify whether users are authenticated and authorized on inbound or outbound connections, or for connections that arrive at a specific interface.
- *local_ip*, *local_mask*—Specify the IP address on the higher security level interface from which or to which access is sought.
- *foreign_ip*, *foreign_mask*—Specify the IP address on the lower security level interface from which or to which access is sought.

Configuring RADIUS Authorization

PIX Firewall now allows a RADIUS server to send user group attributes to the PIX Firewall in the RADIUS authentication response message.

The administrator first defines access lists on the PIX Firewall for each user group. For example, there could be access lists for each department in an organization, sales, marketing, engineering, and so on. The administrator then defines each access list in the group profile in CiscoSecure.

After the PIX Firewall authenticates a user, it can then use the CiscoSecure **acl** attribute returned by the authentication server to identify an access list for a given user group. To maintain consistency, PIX Firewall also provides the same functionality for TACACS+.

To restrict users in a department to three servers and deny everything else, the **access-list** command statements are as follows:

```
access-list eng permit ip any server1 255.255.255.255
access-list eng permit ip any server2 255.255.255.255
access-list eng permit ip any server3 255.255.255.255
access-list eng deny ip any any
```

In this example, the vendor specific attribute string in the CiscoSecure configuration has been set to **acl=eng**. Use this field in the CiscoSecure configuration to identify the **access-list** identification name. The PIX Firewall gets the **acl=***acl_ID* from CiscoSecure and extracts the ACL number from the attribute string, which it puts in a user's uauth entry. When a user tries to open a connection, PIX Firewall checks the access list in the user's uauth entry, and depending on the permit or deny status of the access list match, permits or denies the connection. When a connection is denied, PIX Firewall generates a corresponding syslog message. If there is no match, then the implicit rule is to deny.

Because the source IP of a given user can vary depending on where they are logging in from, set the source address in the **access-list** command statement to **any**, and the destination address to identify which network services the user is permitted or denied access to. If you want to specify that only users logging in from a given subnet may use the specified services, specify the subnet instead of using **any**.

There is *not* a **radius** option to the **aaa authorization** command. You enable RADIUS authorization as follows:

Step 1 Enable RADIUS authentication with the **aaa authentication** command.

Step 2 Create the **access-list** command statements to specify what services hosts are authorized to use with RADIUS.

Step 3 Configure the authentication server with the vendor-specific **acl=***acl_ID* identifier to specify the **access-list** ID.

When the PIX Firewall sends a request to the authentication server, it returns the **acl=***acl_ID* string, which tells PIX Firewall to use the access-list command statements to determine how RADIUS users are authorized.

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 **access-list** commands:

a. For inbound server access, include an **access-list** command statement group for every **static** command you specify.

b. Include an **access-group** command statement for every **access-list** command statement group. Ensure that the ID on the **access-group** command matches those on the **access-list** command statements. You can make **access-list** and **access-group** command statements more readable by using a name for the ID that includes the interface to which you are binding the **access-list** command statements; for example, "acl_dmz1" to bind to the dmz1 interface.

c. Bind only one **access-list** command statement group to an interface.

d. Code **access-list** command statements as tightly as possible. For example, specify which network can access the **access-list** command statement and specify the exact port for which you permit access.

e. Make sure that the global address in the **static** command is the same in the **access-list** 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
access-list acl_dmz1 permit tcp 10.1.1.0 255.255.255.0 host 10.1.1.2 eq smtp
access-group acl_dmz1 in interface dmz1
```

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 **access-list** command statement specifies that only users on the 10.1.1.0 network can access the server via the SMTP port (25).

f. Check that each **static** and **access-list** command statement pair has the correct addresses.

g. Check that two **access-list** 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 **access-list** command statement for TCP and one for UDP. For DNS, if you are only receiving zone updates, you only need a single **access-list** command statement for TCP. Refer to the section "Step 13—Add Inbound Server Access" for an example of two **access-list** command statements for the PPTP protocol.

Step 6 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 7 Ensure that there is only one default **route** command statement to the outside interface.

Step 8 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 9 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 10 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.
- Use the **write memory** command often to save your configuration to Flash memory.
- Use the **write memory** and **reload** commands after changing **alias**, **access-list**, **global**, **nat**, or **static** commands.
- Use the **no failover** command to disable failover if it is not in use. The PIX 506 does not support failover.
- Make sure the MTU is no more than 1500 bytes for Ethernet, or 8192 for either Token Ring or FDDI.