



## Basic Firewall Configuration

---

This chapter describes the basic preparation and configuration required to use the network firewall features of the Cisco PIX Firewall. After completing this chapter, you will be able to establish basic connectivity from your internal network to the public Internet or resources on your perimeter network. The basic configuration described in this chapter lets protected network users start connections, but prevents users on unprotected networks from accessing (or attacking) protected hosts.

This chapter contains the following sections:

- [Using the Command-Line Interface](#)
- [Getting Ready to Configure the PIX Firewall](#)
- [Configuring PIX Firewall Interfaces](#)
- [Configuring the PIX Firewall for Routing](#)
- [Establishing Outbound Connectivity with NAT and PAT](#)
- [Testing Connectivity](#)
- [Saving Your Configuration](#)
- [Troubleshooting](#)
- [Basic Configuration Example for Two Interfaces](#)
- [Basic Configuration Example for Three Interfaces](#)

The last two sections contain examples demonstrating how to configure a PIX Firewall with or without NAT and PAT.

### Using the Command-Line Interface

This section includes the following topics, which describe how to use the PIX Firewall command-line interface (CLI):

- [Access Modes](#)
- [Accessing Configuration Mode](#)
- [Abbreviating Commands](#)
- [Backups](#)
- [Command Line Editing](#)
- [Command Output Paging](#)
- [Comments](#)

- [Configuration Size](#)
- [Help Information](#)
- [Viewing the Default Configuration](#)

**Note**

The PIX Firewall CLI uses similar syntax and other conventions to the Cisco IOS CLI, but the PIX Firewall operating system is not a version of Cisco IOS software. Do not assume that a Cisco IOS CLI command works or has the same function with the PIX Firewall.

## Access Modes

The PIX Firewall provides three administrative access modes:

- Unprivileged mode is available when you first access the PIX Firewall and displays the “>” prompt. This mode lets you view restricted settings.
- Privileged mode displays the “#” prompt and lets you change current settings. Any unprivileged command also works in privileged mode. Use the **enable** command to start privileged mode and the **disable**, **exit**, or **quit** commands to exit.
- Configuration mode displays the “(config)#” prompt and lets you change system configurations. All privileged, unprivileged, and configuration commands work in this mode. Use the **configure terminal** command to start configuration mode and the **exit** or **quit** commands to exit.

## Accessing Configuration Mode

Perform the following steps to access the PIX Firewall Configuration mode.

- 
- 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 506, PIX 515, PIX 525, or PIX 535 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 displays 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 `configure terminal` and press **Enter**. You are now in configuration mode.

---

## Abbreviating Commands

You can abbreviate most commands down to the fewest unique characters for a command; for example, you can enter **wr t** to view the configuration instead of entering the full command **write terminal**, or you can enter **en** to start privileged mode and **con te** to start configuration mode.

In addition, you can enter **0** to represent **0.0.0.0**.

## Backups

You should back up your configuration in at least one of the following ways:

- Store the configuration in Flash memory with the **write memory** command. Should the need arise, you can restore a configuration from Flash memory using the **configure memory** command.
- Use the **write terminal** command to list the configuration. Then cut and paste the configuration into a text file. Then archive the text file. You can restore a configuration from a text file using the **configure terminal** command and pasting the configuration either line by line or as a whole.
- Store the configuration on another system using the **tftp-server** command to initially specify a host and the **write net** command to store the configuration.
- If you have a PIX 520 or older model, store the configuration on a diskette using the **write floppy** command. If you are using Windows, make sure the diskette is IBM formatted. If you are formatting a disk, access the MS-DOS command prompt and use the **format** command. Do not back up your configuration to the PIX Firewall boot disk.

Each image you store overwrites the last stored image.

Should the need arise, you can restore your configuration from Flash memory with the **configure memory** command, or from diskette with the **configure floppy** command.

## Command Line Editing

PIX Firewall uses the same command line editing conventions as Cisco IOS software. You can view all previously entered commands with the **show history** command or individually with the up arrow or **^p** command. Once you have examined a previously entered command, you can move forward in the list with the down arrow or **^n** command. When you reach a command you wish to reuse, you can edit it or press the **Enter** key to start it. You can also delete the word to the left of the cursor with **^w**, or erase the line with **^u**.

PIX Firewall permits up to 512 characters in a command; additional characters are ignored.

## Command Output Paging

On commands such as **help** or **?**, **show**, **show xlate**, or other commands that provide long listings, you can determine if the information displays a screenful and pauses, or lets the command run to completion. The **pager** command lets you choose the number of lines to display before the More prompt appears.

When paging is enabled, the following prompt appears:

```
<--- More --->
```

The More prompt uses syntax similar to the UNIX **more** command:

- To view another screenful, press the Space bar.
- To view the next line, press the **Enter** key.
- To return to the command line, press the **q** key.

## Comments

You can precede a line with a colon (:) to create a comment. However, the comment only appears in the command history buffer and not in the configuration. Therefore, you can view the comment with the **show history** command or by pressing an arrow key to retrieve a previous command, but because the comment is not in the configuration, the **write terminal** command does not display it.

## Configuration Size

The maximum size of a configuration is 350 KB. This is true for the PIX 515, the PIX 520, and any previous PIX Firewall models. Use the UNIX **wc** command or a Windows word processing program, such as Microsoft Word, to view the number of characters in the configuration.

## Help Information

Help information is available from the PIX Firewall command line by entering **help** or a question mark to list all commands, or after a command to list command syntax; for example, **arp ?**.

The number of commands listed when you use the question mark or **help** command differs by access mode so that unprivileged mode offers the least commands and configuration mode offers the greatest number of commands.

In addition, you can enter any command by itself on the command line and then press **Enter** to view the command syntax.

## Viewing the Default Configuration

When you start your PIX Firewall for the first time, the configuration comes with many of the commands you need to get started. The configuration you first receive is known as the default configuration. You can use the **write terminal** command to view your configuration at any time. Also use the **write memory** command frequently to save your configuration to Flash memory.

# Getting Ready to Configure the PIX Firewall

This section describes what you should know and what you should have completed before starting the basic configuration of your PIX Firewall. It contains the following topics:

- [Developing a Security Policy](#)
- [Planning Your Implementation](#)
- [Setting the Default Route for Network Routers](#)
- [Setting the Default Route for Network Hosts](#)

## Developing a Security Policy

The key to successful implementation of your PIX Firewall is having a clear security policy that describes how to control access and use of your organization's network resources. You need to understand your security policy to ensure that you implement and configure the PIX Firewall in a way that supports this policy. Your security policy should have the support of the various departments and administrators responsible for its implementation and should be well understood by network users.

## Planning Your Implementation

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.

You can use the configuration forms in [Appendix A, "Firewall Configuration Forms,"](#) to help you plan your implementation and to collect the information required. The following are the first four configuration forms in the appendix, which will help you collect the information required to complete the configuration described in this chapter:

- PIX Firewall Network Interface Information
- Routing Information
- Network Address Translation
- Static Address Translation

## Setting the Default Route for Network Routers

A router discovers and stores the paths through the network, known as routes. When a router does not have a route to the destination address in a specific packet, it forwards the packet using a default route to another router, called the default router. Configure the default routes on your routers to forward traffic to the PIX Firewall by completing the following steps.

- 
- Step 1** Telnet to the router that connects to the inside interface of the PIX Firewall, or connect to the router's console port.
- If you are using a Windows PC, you can connect to the console port using the HyperTerminal program. You will need to know the username and password for the router.
- Step 2** Access the Cisco IOS configuration mode.

- Step 3** Set the default route to the inside interface of the PIX Firewall with the following Cisco IOS CLI command:
- ```
ip route 0.0.0.0 0.0.0.0 pix_inside_interface_ip_address
```
- Step 4** Enter the **show ip route** command and make sure that the connected PIX Firewall interface is listed as the “gateway of last resort.”
- Step 5** Clear the ARP cache with the **clear arp** command. Then enter **ctrl-z** to exit configuration mode.
- Step 6** From the router, if you changed the default route, use the **write memory** command to store the configuration in Flash memory.
- Step 7** Connect to other routers on the inside and each perimeter interface of the PIX Firewall and repeat Steps 1 through 6 for each router.
- Step 8** 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 the Default Route for Network Hosts

Each host on the same subnet as the inside or perimeter interfaces should have its default route pointing to the PIX Firewall. [Table 2-1](#) summarizes how to set a default route for different types of hosts:

**Table 2-1** *Setting the Default Route for Different Network Hosts*

Host Type	To Change the Default Route	To View the Default Route
Solaris or SunOS	<ol style="list-style-type: none"> <li>1. With root permissions, edit the <code>/etc/defaultrouter</code> file to point the default route at the PIX Firewall.</li> <li>2. Reboot the workstation.</li> </ol>	Enter the following command: <b>netstat -nr</b>
LINUX	With root permissions, enter the following command: <b>route add default gw IP_address_of_next_host</b>	Enter the following command: <b>netstat -nr</b>
Windows 95, Windows 98, and Windows 2000	<ol style="list-style-type: none"> <li>1. Click <b>Start&gt;Settings&gt;Control Panel</b> and double-click the <b>Network</b> item.</li> <li>2. Select the TCP/IP entry from the list of installed network components and click <b>Properties</b>.</li> <li>3. Click the <b>Gateway</b> tab to set the default.</li> </ol>	Select <b>Start&gt;Run</b> and enter the following command: <b>winipcfg</b>

**Table 2-1** Setting the Default Route for Different Network Hosts (continued)

Host Type	To Change the Default Route	To View the Default Route
Windows NT	<ol style="list-style-type: none"> <li>1. Click the <b>Protocols</b> tab on the Network control panel.</li> <li>2. In the Network Protocols window, click <b>TCP/IP Protocol</b>, and click <b>Properties</b>.</li> <li>3. In the Microsoft TCP/IP Properties window, click the <b>IP Address</b> tab.</li> <li>4. Click <b>Advanced</b> and click <b>Remove</b>.</li> <li>5. Click <b>Add</b> and enter the IP address for the PIX Firewall interface.</li> <li>6. Close each window and click <b>Yes</b> when you are prompted to restart Windows.</li> </ol>	From the command prompt, enter the following command:  <b>ipconfig</b>
MacOS 7.5 and later	From the <b>Apple</b> menu, select <b>Control Panels&gt;TCP/IP</b> window	From the <b>Apple</b> menu, select <b>Control Panels&gt;TCP/IP</b> window

## Configuring PIX Firewall Interfaces

This section includes the following topics, which describe the configuration required for each PIX Firewall interface:

- [Assigning an IP Address and Subnet Mask](#)
- [Identifying the Interface Type](#)
- [Changing Interface Names or Security Levels](#)

### Assigning an IP Address and Subnet Mask

Assign an **ip address** command to each interface in your PIX Firewall that connects to another 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 should 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.

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
```

## Identifying the Interface Type

All interfaces in a new PIX Firewall are shut down by default. You need to use the **interface** command to explicitly enable each interface you are using.

If you have Ethernet interfaces in the PIX Firewall, the default configuration provides the necessary options for the **interface** command. If your PIX Firewall has gigabit Ethernet refer to the **interface** command page in the *Cisco PIX Firewall Command Reference* for configuration information.

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

```
interface hardware_id hardware_speed [shutdown]
```

- Replace *hardware\_id* with the hardware name for the network interface card, such as **ethernet2** and **ethernet3**, and so forth. You can abbreviate the *hardware\_id* name with any significant letters, such as, **e0** for **ethernet0**. If one of the Ethernet cards is a 4-port card, the Ethernet names change to correspond to the slot in which the card resides.
- Replace *hardware\_speed* with the speed of the interface, using the values shown in [Table 2-2](#).

The **shutdown** option disables use of the interface. When you first install PIX Firewall, all interfaces have the **shutdown** option in effect.

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.

**Table 2-2 Values for the hardware\_speed Parameter**

Value	Description
<b>10baset</b>	10 Mbps Ethernet half-duplex communications.
<b>100basetx</b>	100 Mbps Ethernet half-duplex communications.
<b>100full</b>	100 Mbps Ethernet full-duplex communications.

**Table 2-2 Values for the hardware\_speed Parameter (continued)**

Value	Description
<b>1000sxfull</b>	1000 Mbps Gigabit Ethernet full-duplex operation.
<b>1000basesx</b>	1000 Mbps Gigabit Ethernet half-duplex operation.
<b>1000auto</b>	1000 Mbps Gigabit Ethernet to auto-negotiate full or half duplex.
<b>au</b>	10 Mbps Ethernet half-duplex communications for an AUI cable interface.
<b>auto</b>	Sets Ethernet speed automatically. We recommend that you not use this setting to ensure compatibility with switches and routers in your network.
<b>bnc</b>	10 Mbps Ethernet half-duplex communications for a BNC cable interface.

**Note**

Make sure the MTU is no more than 1500 bytes for Ethernet.

## Changing Interface Names or Security Levels

Each interface has a unique name and security level that you can change using the **nameif** command. By default, Ethernet0 is named outside and assigned the level security0. Ethernet1 is named inside with the level security 100. By default, perimeter interfaces are named *infn*, where *n* represents the position of the interface card in the PIX Firewall. The default security level of perimeter interfaces starts at security10 for ethernet2 (intf2), and increments by 5 for each additional interface.

**Note**

You can skip this section if you are using the default interface names and security levels.

Use the show **nameif** command to view the current names and security levels for each interface. The results of this command for a PIX Firewall with three interfaces might be as follows.

```
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 intf2 security10
```

Security levels let you control access between systems on different interfaces and the way you enable or restrict access depends on the relative security level of the interfaces:

- To enable access to a higher security level interface from a lower level interface—use the **static** and **access-list** commands
- To enable access to a lower level interface from a higher level interface—use the **nat** and **global** commands

Locate servers on the lowest security level perimeter interface, because if compromised, the attacker could only easily attack an interface with a lower security level, the outside. The only exception to putting servers on the lowest perimeter interface is the TFTP server where you download configurations from

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

```
nameif hardware_id interface security_level
```

- Replace *hardware\_id* with the value used in the **interface** command, such as **ethernet0**.
- Replace *interface* with any meaningful name, such as **dmz** or **perim** for each perimeter interface.  
You will need to enter this name frequently, so a shorter name is a better choice, although you can use up to 48 characters. The default names are *inftn*, where *n* represents the position of the interface card in the PIX Firewall.
- Replace *security\_level* with a value such as **security40** or **security60**.  
The default security levels for perimeter interfaces increment by 5 for each interface starting with **security10** for *inft2* (the default name for the first perimeter interface). For example, *inft3* = **security15**, *inft4* = **security 20**, and *inft5* = **security 25**.  
You can choose any unique security level between 1 and 99 for a perimeter interface. If you have four or more interfaces, it will be easier to enter your configuration if you use the higher security level for the perimeter interface with more hosts.

## Configuring the PIX Firewall for Routing

Each inside or perimeter PIX Firewall interface is configurable for route and Routing Information Protocol (RIP) information. To determine what route information is required, consider what routers are in use in your network and are adjacent to the planned installation point of the PIX Firewall.

Specifying a route tells the PIX Firewall where to send information that is forwarded on a specific interface and destined for a particular network address. You can specify more than one route per interface, allowing you control where to send network traffic. Refer to the **route** command page in [Cisco PIX Firewall Command Reference](#) for more information.

The PIX Firewall learns where everything is on the network by “passively” listening for RIP network traffic. When the PIX Firewall interface receives RIP traffic, the PIX Firewall updates its routing tables. You can also configure the PIX Firewall to broadcast an inside or perimeter interface as a “default” route. Broadcasting an interface as a default route is useful if you want all network traffic on that interface to go out through that interface. Refer to the **rip** command page in [Cisco PIX Firewall Command Reference](#) for configuration information.

When defining a route, specify the IP address and network mask for the destination network. Use 0.0.0.0 for both the IP address and network mask as the default value.

The gateway IP address is the router that routes the traffic to the destination network IP address.

RIP configuration specifies whether the PIX Firewall updates its routing tables by passive listening to RIP traffic, and whether the interface broadcasts itself as a default route for network traffic on that interface. If you configure the PIX Firewall interface to listen for RIP updates, be sure to configure the router supplying the RIP information with the network address for the PIX Firewall interface.



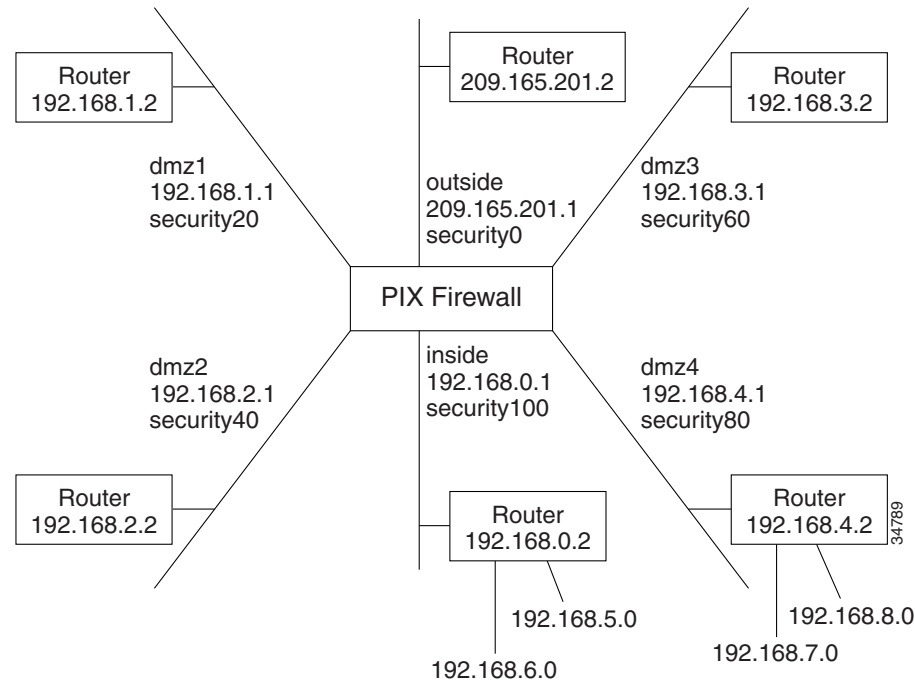
### Note

Before testing your configuration, flush the ARP caches on any routers that feed traffic into or from the PIX Firewall and between the PIX Firewall and the Internet. For Cisco routers, use the **clear arp** command to flush the ARP cache.

Follow these steps to add static routes:

- Step 1** Sketch out a diagram of your network as shown in [Figure 2-1](#).

**Figure 2-1 Sketch Network with Routes**



- Step 2** Enter the default route:

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

Only one default route is permitted. 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, 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.

---

## Establishing Outbound Connectivity with NAT and PAT

Mapping a range of global IP addresses to an inside or perimeter address, or to a set of addresses, is known as Network Address Translation (NAT). Mapping a single global IP address to many inside or perimeter addresses is known as Port Address Translation (PAT). PAT extends the range of available outside addresses at your site by dynamically assigning unique port numbers to the outside address as a connection is requested. A single IP address has up to 64,000 ports that are available for making connections. For PAT, the port number uniquely identifies each connection.

The PIX Firewall associates internal addresses with global addresses using a NAT identifier (NAT ID). For example, if the inside interface has NAT ID5, then hosts making connections from the inside interface to another interface (perimeter or outside) get a substitute (translated) address from the pool of global addresses associated with NAT ID5.

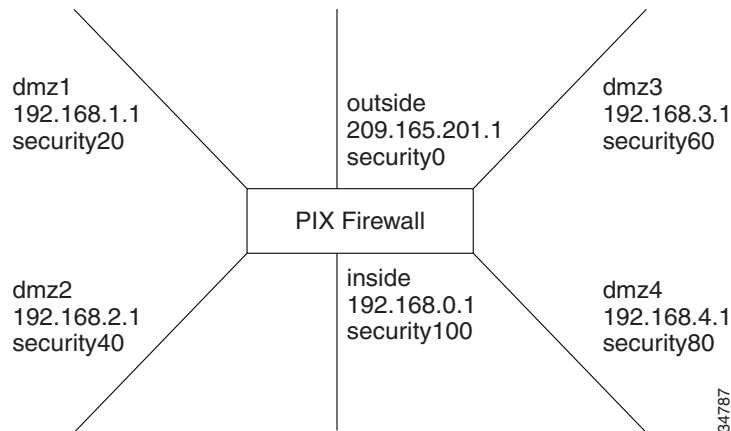
If you decide not to use NAT to protect internal addresses from exposure on outside networks, assign those addresses NAT ID 0, which indicates to the PIX Firewall that translation is not provided for those addresses. Refer to *Cisco PIX Firewall Command Reference* for configuration information.

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 *Cisco PIX Firewall Command Reference*.

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.

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 2-2](#).

**Figure 2-2 Sketching Interfaces and Security Levels**

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

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

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

For example, to let only host 192.168.2.42 start connections on the dmz2 interface, you could specify 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 has to 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 should 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 entry, which permits up to 64,000 hosts to use a single IP address. PAT has some restrictions in its use such as it cannot support H.323 or caching nameserver use, so you may want to use it to augment a range of global addresses rather than using it as your sole global address.

For example:

```
global (outside) 1 209.165.201.5 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.

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 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.

# Testing Connectivity

You can use the **access-list** command to ping from a host on an interface through the PIX Firewall to a host on another interface. This allows you to 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.



## Note

We recommend that you only enable pinging during troubleshooting.

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

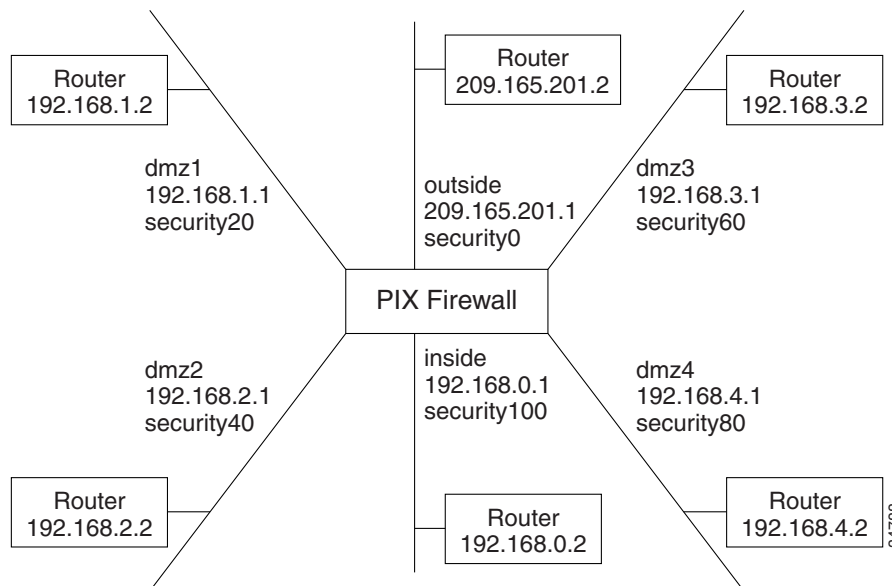
```
no access-list acl_in permit icmp any any
no access-list acl_out permit icmp any any
no access-list acl_dmz1 permit icmp any any
```

You can also remove the **access-group** command statements, but be sure not to remove those associated with other **access-list** command statements. To test your connectivity, perform the following steps:

**Step 1** Start with a sketch of your PIX Firewall, with each interface connected to the inside, outside, and any perimeter networks.

Figure 2-3 shows an example:

**Figure 2-3 Sketch a Network with Interfaces and Routers**



**Step 2** Enable Pinging.

Enter 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, enter the following command statement to ping from the inside interface to the outside interface:

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

**Step 3** Enable debugging.

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 4** 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 software version 5.3 and later, 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 the following:

- 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.
- 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."
- 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.

---

## Saving Your Configuration

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 basic configuration lets protected network users start connections, but prevents users on unprotected networks from accessing (or attacking) protected hosts.

Use the **write terminal** command to view your current configuration.

## Troubleshooting

Perform the following steps to ensure that your configuration is correct.

- 
- Step 1** Make sure that each interface you intend to operate has the **shutdown** option disabled. Refer to [“Configuring PIX Firewall Interfaces”](#) 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** 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. 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 should 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 follows:

```
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.

- a. At the PIX Firewall CLI prompt, enter 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 PIX Firewall is correctly configured and a cable is attached.

- b. Check that network cables are attached.

- Step 10 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 11 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 so that you can ping from the lower security level interfaces to the higher security level interfaces.

## Basic Configuration Example for Two Interfaces

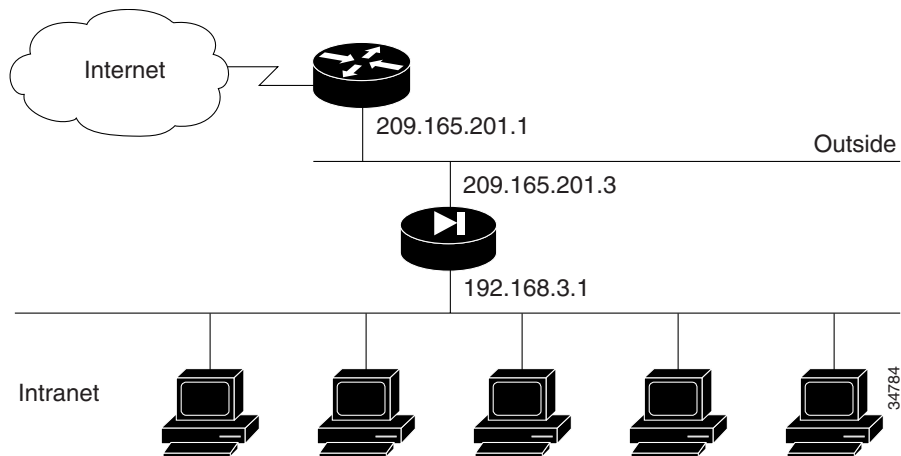
This section contains basic configuration examples for configuring the PIX Firewall with two interfaces. It contains the following topics:

- [Two Interfaces Without NAT or PAT](#)
- [Two Interfaces with NAT and PAT](#)

### Two Interfaces Without NAT or PAT

When you first add a PIX Firewall to an existing network, it is easiest to implement its use if you do not have to renumber all the inside and outside IP addresses. The configuration in [Figure 2-4](#) illustrates this scenario. All inside hosts can start connections. All external hosts are blocked from initiating connections or sessions on inside hosts.

**Figure 2-4** Two Interfaces Without NAT



The values given are examples only. You should change this configuration for the information and requirements that are specific for your network.

The following steps describe the configuration procedure that is the same regardless of how you implement your PIX Firewall:

---

**Step 1** Identify the security level and names of each interface by entering the following commands:

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

**Step 2** Identify the line speed of each interface by entering the following commands:

```
interface ethernet0 10baset
interface ethernet1 10baset
```

You may get better performance by changing the default **auto** option in the **interface** command to the specific line speed for the interface card.

**Step 3** Identify the IP addresses for each interface:

```
ip address outside 209.165.201.3 255.255.255.224
```

```
ip address inside 209.165.200.225 255.255.255.0
```

- Step 4** Specify the host name for the PIX Firewall:

```
hostname pixfirewall
```

This name appears in the command line prompt.

- Step 5** Set the ARP timeout to 14,400 seconds (four hours):

```
arp timeout 14400
```

With this command, entries are kept in the ARP table for four hours before they are flushed. Four hours is the standard default value for ARP timeouts.

- Step 6** Disable failover access:

```
no failover
```

- Step 7** Enable the use of text strings instead of IP addresses:

```
names
```

This makes your configuration files more readable.

- Step 8** Enable paging:

```
pager lines 24
```

When 24 lines of information display, PIX Firewall pauses the listing and prompts you to continue.

- Step 9** Enable syslog messages, which provide diagnostic information and status for the PIX Firewall:

```
logging buffered debugging
```

PIX Firewall makes it easy to view syslog messages with the show logging command.

- Step 10** Let inside IP addresses be recognized on the outside network and let inside users start outbound connections:

```
nat (inside) 0 209.165.200.225 255.255.255.0
```

- Step 11** Set the outside default route to the router attached to the Internet:

```
route outside 0.0.0.0 0.0.0.0 209.165.201.1 1
```

- Step 12** Allow inbound and outbound pings:

```
access-list ping_acl permit icmp any any
access-group ping_acl in interface inside
access-group ping_acl in interface dmz
access-list acl_out permit icmp any any
```

The “ping\_acl” **access-list** command statement group is bound to the inside interface. The “acl\_out” group is bound to the outside interface. This distinction accommodates the **access-list** command statement later in the configuration that applies permissions to a **static** command statement mapping. When troubleshooting is complete, remove the ICMP **access-list** statements.

- Step 13** Set the default values for the maximum duration that PIX Firewall resources can remain idle until being freed:

```
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00
udp 0:02:00 rpc 0:10:00 h323 0:05:00
sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
```

Additional users cannot make connections until a connection resource is freed either by a user dropping a connection or by an xlate and conn timer time out.

**Step 14** Disable SNMP access and SNMP traps generation:

```
no snmp-server location
no snmp-server contact
snmp-server community public
```

**Step 15** Set the maximum transmission unit value for Ethernet access:

```
mtu outside 1500
mtu inside 1500
```

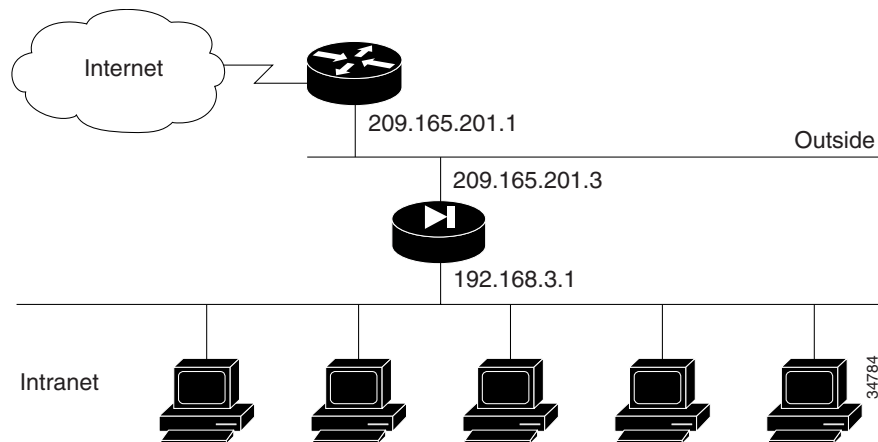
[Example 2-1](#) shows the listing for the basic configuration required to implement a PIX Firewall with two interfaces without NAT.

### **Example 2-1 Two Interfaces Without NAT**

```
nameif ethernet0 outside security0
nameif ethernet1 inside security100
interface ethernet0 10baset
interface ethernet1 10baset
ip address outside 209.165.201.3 255.255.255.224
ip address inside 209.165.200.225 255.255.255.0
hostname pixfirewall
arp timeout 14400
no failover
names
pager lines 24
logging buffered debugging
nat (inside) 0 209.165.200.225 255.255.255.0
route outside 0.0.0.0 0.0.0.0 209.165.201.1 1
access-list ping_acl permit icmp any any
access-group ping_acl in interface inside
access-group ping_acl in interface dmz
access-list acl_out permit icmp any any
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00
udp 0:02:00 rpc 0:10:00 h323 0:05:00
sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
no snmp-server location
no snmp-server contact
snmp-server community public
mtu outside 1500
mtu inside 1500
```

## Two Interfaces with NAT and PAT

Use NAT if the network addresses in use on your internal network are not valid for use on the public Internet, or when you want to hide your network addresses from potential attackers. Use PAT when you do not have a large enough pool of registered IP addresses for all the users on your internal network that require concurrent connectivity to the public Internet. [Figure 2-5](#) illustrates a network using unregistered IP addresses on the intranet, which requires NAT for connecting to the public Internet.

**Figure 2-5 Two Interfaces With NAT**

The following steps show how to change the example given in “[Two Interfaces Without NAT or PAT](#)” for enabling NAT and PAT:

**Step 1** Identify the IP addresses for each interface:

```
ip address outside 209.165.201.3 255.255.255.224
ip address inside 192.168.3.0 255.255.255.0
```

This step differs from “[Two Interfaces Without NAT or PAT](#)” because the inside IP addresses in this example are unregistered.

**Step 2** Enter the following command to enable NAT and PAT:

```
nat (inside) 1 0 0
```

This permits all inside users to start outbound connections using the translated IP addresses from a global pool. This command replaces the command in [Step 10](#) in “[Two Interfaces Without NAT or PAT](#).”

**Step 3** Create a pool of global addresses that translated addresses use when they exit the PIX Firewall from the protected networks to the unprotected networks:

```
global (outside) 1 209.165.201.10-209.165.201.30
global (outside) 1 209.165.201.8
```

The **global** command statement is associated with a **nat** command statement by the NAT ID, which in this example is 1. Because there are limited IP addresses in the pool, a PAT external (global) address is added to handle overflow.

[Example 2-2](#) shows the complete configuration for configuring two interfaces with NAT.

#### **Example 2-2 Two Interfaces with NAT**

```
nameif ethernet0 outside security0
nameif ethernet1 inside security100
interface ethernet0 10baset
interface ethernet1 10baset
ip address outside 209.165.201.3 255.255.255.224
ip address inside 192.168.3.0 255.255.255.0
hostname pixfirewall
arp timeout 14400
```

```
no failover
names
pager lines 24
logging buffered debugging
nat (inside) 1 0 0
global (outside) 1 209.165.201.10-209.165.201.30
global (outside) 1 209.165.201.8
route outside 0.0.0.0 0.0.0.0 209.165.201.1 1
access-list acl_out permit icmp any any echo-reply
access-list acl_out permit icmp any any unreachable
access-list acl_out permit icmp any any time-exceeded
access-group acl_out in interface outside
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00
udp 0:02:00 rpc 0:10:00 h323 0:05:00
sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
no snmp-server location
no snmp-server contact
snmp-server community public
mtu outside 1500
mtu inside 1500
```

## Basic Configuration Example for Three Interfaces

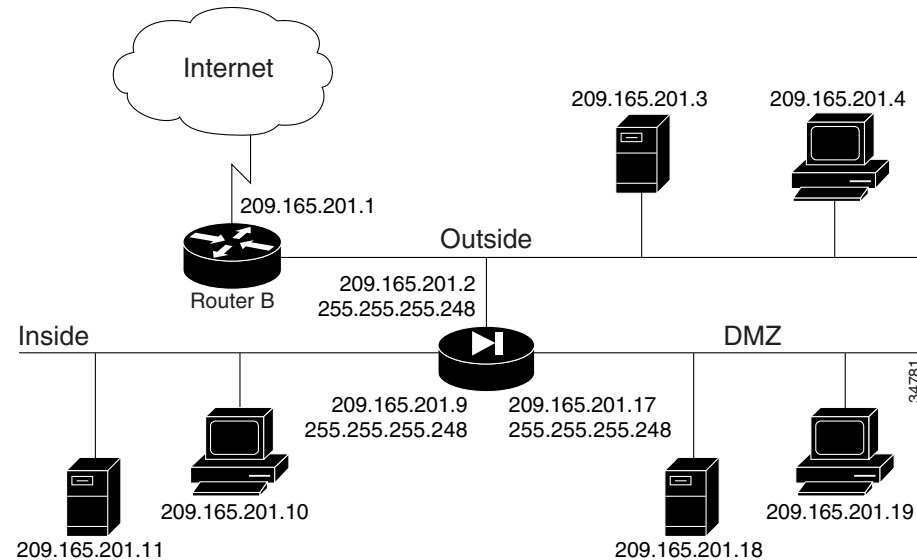
This section contains the following topics:

- [Three Interfaces Without NAT or PAT](#)
- [Three Interfaces with NAT and PAT](#)

## Three Interfaces Without NAT or PAT

In [Figure 2-6](#), the PIX Firewall has three interfaces configured without address translation.

**Figure 2-6 Three-interface Configuration**



The network has the following IP addresses and network masks:

- Outside network interface address: 209.165.201.2, network mask: 255.255.255.248
- Inside network interface address: 209.165.201.9, network mask: 255.255.255.248
- DMZ network interface address: 209.165.201.17, network mask: 255.255.255.248

The following procedure shows the way the configuration for this example differs from the example shown in [“Two Interfaces Without NAT or PAT.”](#)

**Step 1** Identify the security level and names of each interface by entering the following commands:

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

An additional **nameif** command is required for the third interface in this example.

**Step 2** Identify the line speed of each interface by entering the following commands:

```
interface ethernet0 10baset
interface ethernet1 10baset
interface ethernet2 100basetx
```

An additional **interface** command is required for the third interface in this example.

**Step 3** Identify the IP addresses for each interface:

```
ip address outside 209.165.201.3 255.255.255.224
ip address inside 172.31.2.100 255.255.255.0
ip address dmz 209.165.201.17 255.255.255.248
```

An additional IP address is required for the third interface in this example.

**Step 4** Map access to the 209.165.201.19 host on the dmz interface:

```
static (dmz,outside) 209.165.201.19 209.165.201.19 netmask 255.255.255.248
```

**Step 5** Use the **access-list** command to let any outside user access the DMZ host on any port:

```
access-list acl_out permit tcp any host 209.165.201.19  
access-group acl_out in interface outside
```

The **access-list** command lets any outside user access the host on any port.

[Example 2-3](#) shows the complete configuration for three interfaces without NAT. The commands that are different from the example shown in “[Two Interfaces Without NAT or PAT](#),” are shown in boldface type.

### **Example 2-3 Three Interfaces Without NAT**

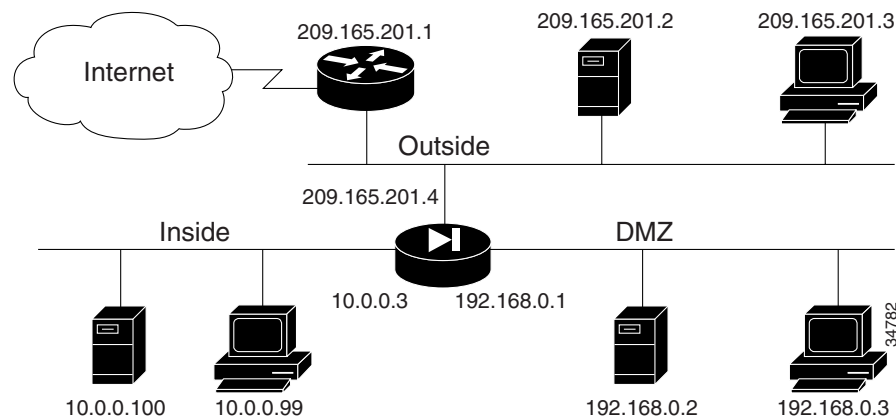
```
nameif ethernet0 outside security0  
nameif ethernet1 inside security100  
nameif ethernet2 dmz security50  
interface ethernet0 10baset  
interface ethernet1 10baset  
interface ethernet0 100basetx  
ip address outside 209.165.201.3 255.255.255.224  
ip address inside 172.31.2.100 255.255.255.0  
ip address dmz 209.165.201.17 255.255.255.248  
hostname pixfirewall  
arp timeout 14400  
no failover  
names  
pager lines 24  
logging buffered debugging  
nat (inside) 0 172.31.2.0 255.255.255.0  
static (dmz,outside) 209.165.201.19 209.165.201.19 netmask 255.255.255.248  
access-list acl_out permit tcp any host 209.165.201.19  
access-group acl_out in interface outside  
route outside 0.0.0.0 0.0.0.0 209.165.201.1 1  
access-list ping_acl permit icmp any any  
access-group ping_acl in interface inside  
access-group ping_acl in interface dmz  
access-list acl_out permit icmp any any  
timeout xlate 3:00:00  
timeout conn 1:00:00 half-closed 0:10:00  
udp 0:02:00 rpc 0:10:00 h323 0:05:00  
sip 0:30:00 sip_media 0:02:00  
timeout uauth 0:05:00 absolute  
no snmp-server location  
no snmp-server contact  
snmp-server community public  
mtu outside 1500  
mtu inside 1500
```

## Three Interfaces with NAT and PAT

In [Figure 2-7](#), the PIX Firewall has three interfaces and these attributes:

- Address translation is performed between the interfaces.
- A web server on the DMZ interface is publicly accessible. The **name** command maps its host address to the name “webservr.”
- The inside network has illegal addresses (10.0.0.0), the DMZ interface has RFC 1597 addresses (192.168.0.0), and the outside network has legal, registered addresses (209.165.201.0).
- TCP and UDP connections from the inside are allowed to go out on the DMZ and outside.
- An inside host has been given Telnet access to the PIX Firewall console.

**Figure 2-7** Three Interfaces with NAT



The network has the following IP addresses and network masks:

- Outside network interface address: 209.165.201.4, network mask: 255.255.255.224
- Allowable global and static addresses on the outside network: 209.165.201.5-209.165.201.30, network mask: 255.255.255.224
- Inside network interface address: 10.0.0.3, network mask: 255.0.0.0
- DMZ network interface address: 192.168.0.1, network mask: 255.255.255.0

The following procedure shows the commands that differ from the example shown in “[Three Interfaces Without NAT or PAT](#).”

- Step 1** Create a pool of global addresses for the outside and DMZ interfaces. Because there are limited outside IP addresses, add a PAT global to handle overflow. The **global (dmz)** command gives inside users access to the web server on the DMZ interface.

```
global (outside) 1 209.165.201.10-209.165.201.30
global (outside) 1 209.165.201.5
global (dmz) 1 192.168.0.10-192.168.0.20
```

- Step 2** Let inside users start connections on the DMZ and outside interfaces, and let DMZ users start connections on the outside interface:

```
nat (inside) 1 10.0.0.0 255.0.0.0
nat (dmz) 1 192.168.0.0 255.255.255.0
```

**Step 3** Give the IP address of the web server a label:

```
name 192.168.0.2 webserver
```

**Step 4** Let any user on the outside interface access the web server on the DMZ interface:

```
static (dmz,outside) 209.165.201.6 webserver
access-list acl_out permit tcp any host 209.165.201.6 eq 80
access-group acl_out in interface outside
```

The **access-list** command statement is bound to the outside interface by the **access-group** command statement.

[Example 2-4](#) shows the complete configuration for three interfaces with NAT.

#### **Example 2-4 Three Interfaces with NAT**

```
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 dmz security50
interface ethernet0 10full
interface ethernet1 10full
interface ethernet2 10full
ip address outside 209.165.201.4 255.255.255.224
ip address inside 10.0.0.3 255.0.0.0
ip address dmz 192.168.0.1 255.255.255.0
hostname pixfirewall
arp timeout 14400
no failover
names
pager lines 24
logging buffered debugging
no rip inside passive
no rip outside passive
no rip inside default
no rip outside default
route outside 0.0.0.0 0.0.0.0 209.165.201.1 1
access-list ping_acl permit icmp any any
access-group ping_acl in interface inside
access-group ping_acl in interface dmz
access-list acl_out permit icmp any any
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00
udp 0:02:00 rpc 0:10:00 h323 0:05:00
sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
no snmp-server location
no snmp-server contact
snmp-server community public
mtu outside 1500
mtu inside 1500
mtu dmz 1500
telnet 10.0.0.100 255.255.255.255
telnet timeout 15
global (outside) 1 209.165.201.10-209.165.201.30
global (outside) 1 209.165.201.5
global (dmz) 1 192.168.0.10-192.168.0.20
nat (inside) 1 10.0.0.0 255.0.0.0
nat (dmz) 1 192.168.0.0 255.255.255.0
name 192.168.0.2 webserver
```

```
static (dmz,outside) 209.165.201.6 webserver
access-list acl_out permit tcp any host 209.165.201.6 eq 80
access-group acl_out in interface outside
```

