



## Establishing Connectivity

---

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:

- [Setting Default Routes](#)
- [Configuring PIX Firewall Interfaces](#)
- [Configuring the PIX Firewall for Routing](#)
- [Establishing Outbound Connectivity with NAT and PAT](#)
- [Testing Connectivity](#)
- [Saving Your Configuration](#)
- [Configuration Examples](#)
- [Using Outside NAT](#)
- [Enabling Stub Multicast Routing](#)

### Setting Default Routes

This section describes how to set default routes on devices and hosts that communicate with the PIX Firewall. It includes the following topics:

- [Setting Default Routes for Network Routers](#)
- [Setting the Default Route for Network Hosts](#)

#### Setting Default Routes 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:<br><pre>netstat -nr</pre>                            |
| LINUX                                    | With root permissions, enter the following command:<br><pre>route add default gw IP_address_of_next_host</pre>                                                                                                                                                                                                             | Enter the following command:<br><pre>netstat -nr</pre>                            |
| 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> | Click <b>Start&gt;Run</b> and enter the following command:<br><pre>winipcfg</pre> |

**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:<br><br><code>ipconfig</code> |
| MacOS 7.5 and later | From the <b>Apple</b> menu, select <b>Control Panels&gt;TCP/IP</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | From the <b>Apple</b> menu, select <b>Control Panels&gt;TCP/IP</b>                 |

## 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 interface_name ip_address netmask
```

- Replace *interface\_name* with the name assigned to each PIX Firewall interface. By default, the lowest security interface is named **outside**, while the highest security interface is named **inside**.
- 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 associated with 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.     |
| <b>1000sxfull</b> | 1000 Mbps Gigabit Ethernet full-duplex operation. |

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

| Value             | Description                                                                                                                                      |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <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 intf*n*, 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

An attacker who obtains access to an interface can easily attack other interfaces with a lower security level. For this reason, locate public servers on a perimeter interface with the lowest security level. However, the TFTP server from where you download PIX Firewall configurations should be kept on a more secure interface to prevent unauthorized access.

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 *intfn*, 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 *intf2* (the default name for the first perimeter interface). For example, *intf3* = security15, *intf4* = security 20, and *intf5* = 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

This section describes how to configure the PIX Firewall to correctly route traffic to and from adjacent networks. It includes the following topics:

- [Overview](#)
- [Configuring Static Routes on PIX Firewall](#)

### Overview

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, which lets you control where to send network traffic. Refer to the **route** command page in the *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 the *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 as the default value for both the IP address and network mask.

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**

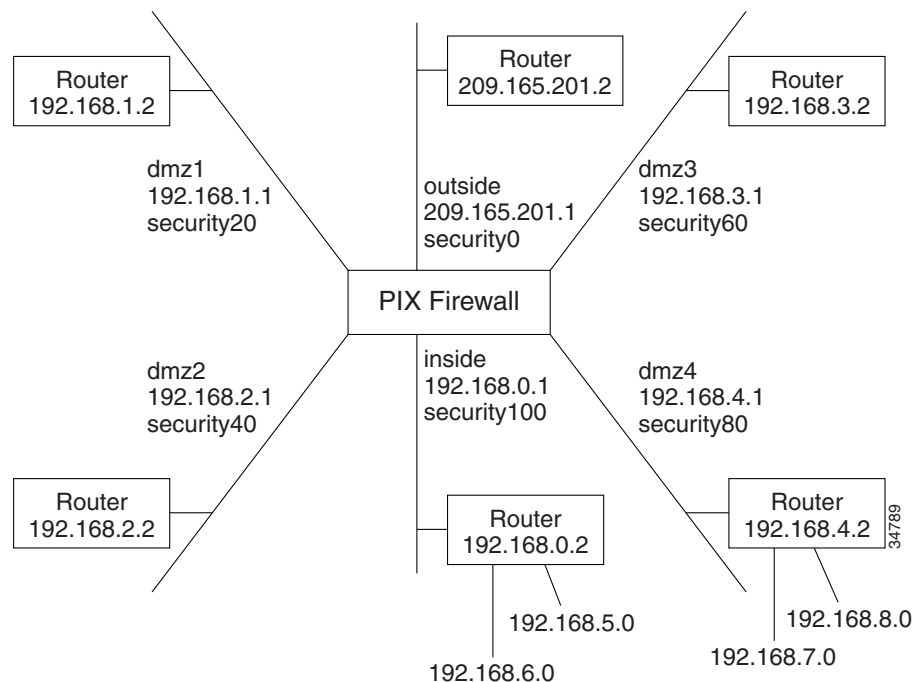
Before testing your configuration, flush the ARP caches on any routers that feed traffic into or from the PIX Firewall and between the PIX Firewall and the Internet. For Cisco routers, use the **clear arp** command to flush the ARP cache.

## Configuring Static Routes on PIX Firewall

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

This section describes how to use Network Address Translation (NAT) and Port Address Translation (PAT) to establish outbound connectivity from hosts on higher security interfaces to hosts on lower security interfaces. It includes the following topics:

- [Overview](#)
- [How NAT and PAT Work](#)
- [Configuring NAT and PAT](#)

### Overview

Network Address Translation (NAT) maps global IP addresses to local IP addresses. Static NAT is described in the “[Allowing Server Access with Static NAT](#)” section in [Chapter 3, “Controlling Network Access and Use.”](#) Static NAT provides a permanent one-to-one map between two addresses. Dynamic NAT uses a range or pool of global addresses to let you support a large number of users with a limited number of global addresses.

Port Address Translation (PAT) maps a single global IP address to many local addresses. 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 65,535 ports that are available for making connections. For PAT, the port number uniquely identifies each connection.

Most often (and always with versions of PIX Firewall earlier than version 6.2) NAT and PAT apply to addresses of inside hosts that are initiating outbound connections through the PIX Firewall. In this case, the global addresses are typically IP addresses registered with the Network Information Center (NIC) for use on the public Internet. The local addresses are internal IP addresses that you do not wish to use on the public Internet. You may wish to translate your internal addresses because they are non-routable (private) or to discourage attacks from the public Internet.

PIX Firewall version 6.2 introduces support for NAT and PAT of addresses on outside networks (lower security interfaces) that initiate connections to hosts on lower security interfaces. Outside NAT is occasionally useful for controlling routing and for connecting networks with overlapping addresses. For more information about Outside NAT, refer to “[Using Outside NAT.](#)”

Table 2-3 summarizes the different functions and applications of NAT and PAT.

**Table 2-3 Address Translation Types**

| Type of Address Translation | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Inside dynamic NAT          | Translates between host addresses on more secure interfaces and a range or pool of IP addresses on a less secure interface. This provides a one-to-one mapping between internal and external address that allows internal users to share registered IP addresses and hides internal addresses from view on the public Internet.                                                                                                                                                                                                                                        |
| Inside dynamic PAT          | Translates between host addresses on more secure interfaces and a single address on a less secure interface. This provides a many-to-one mapping between internal and an external address. This allows internal users to share a single registered IP address and hides internal addresses from view on the public Internet. PAT is supported for fewer applications than is NAT. For restrictions on its use, refer to the “ <a href="#">How Application Inspection Works</a> ” section in Chapter 4, “ <a href="#">Configuring Application Inspection (Fixup)</a> .” |
| Inside static NAT           | Provides a permanent, one-to-one mapping between an IP address on a more secure interface and an IP address on a less secure interface. This allows hosts to access the inside host from the public Internet without exposing the actual IP address.                                                                                                                                                                                                                                                                                                                   |
| Outside dynamic NAT         | Translates between host addresses on less secure interfaces and a range or pool of IP addresses on a more secure interface. This provides a one-to-one mapping between external and internal addresses. This is most useful for controlling the addresses that appear on inside interfaces of the PIX Firewall and for connecting private networks with overlapping addresses.                                                                                                                                                                                         |
| Outside dynamic PAT         | Translates between host addresses on less secure interfaces and a single address on a more secure interface. This provides a many-to-one mapping between external addresses and an internal address.                                                                                                                                                                                                                                                                                                                                                                   |
| Outside static NAT          | Provides a permanent, one-to-one mapping between an IP address on a less secure interface and an IP address on a more secure interface.                                                                                                                                                                                                                                                                                                                                                                                                                                |

## How NAT and PAT Work

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 the *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 the *Cisco PIX Firewall Command Reference*.

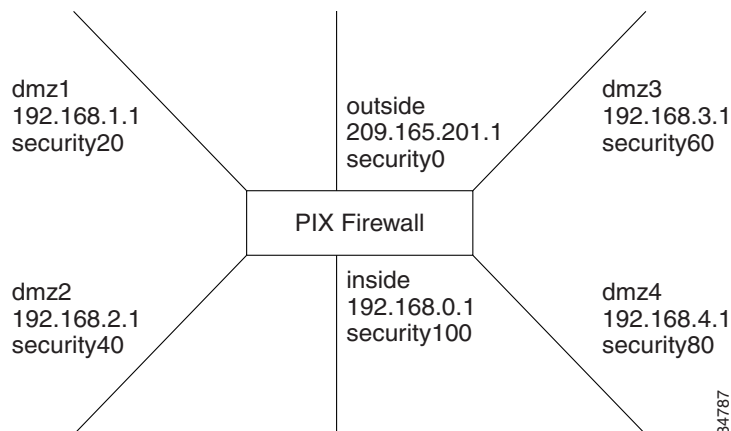
As you enter the **nat** and **global** commands to let users start connections, you can use the **show nat** or **show global** commands to list the existing commands. If you make a mistake, remove the old command with the **no** form of the command, specifying all the options of the first command.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

## Configuring NAT and PAT

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.



**Note** PIX Firewall version 5.2 and higher permits multiple PAT global command statements 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 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.



### 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
```

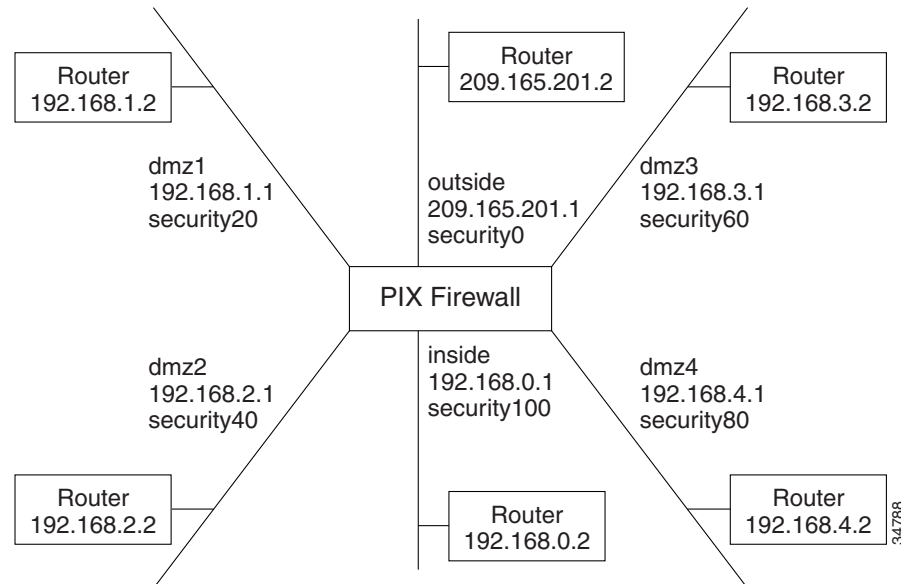
You can also remove the **access-group** command statements, but be sure not to remove those associated with other **access-list** command statements. To test your connectivity, perform the following steps:

---

- 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 you 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.

## Configuration Examples

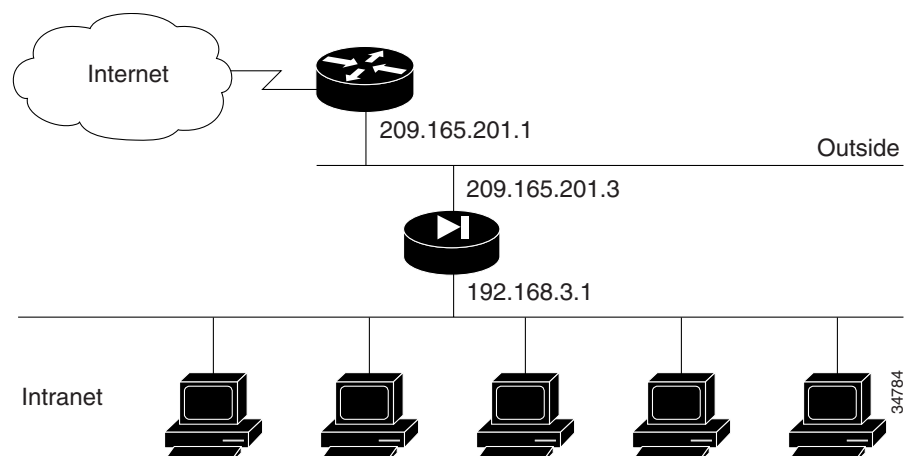
This section illustrates and describes a number of common ways to implement the PIX Firewall. It includes the following topics:

- [Two Interfaces Without NAT or PAT](#)
- [Two Interfaces with NAT and PAT](#)
- [Three Interfaces Without NAT or PAT](#)
- [Three 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 acl_out permit icmp any any
access-group acl_out in interface outside
```

These statements allow the PIX Firewall to forward ICMP replies received on the outside interface. These replies are received in response to ping commands issued from the internal network.




---

**Note** When troubleshooting is complete, remove these 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 acl_out permit icmp any any
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
```

```

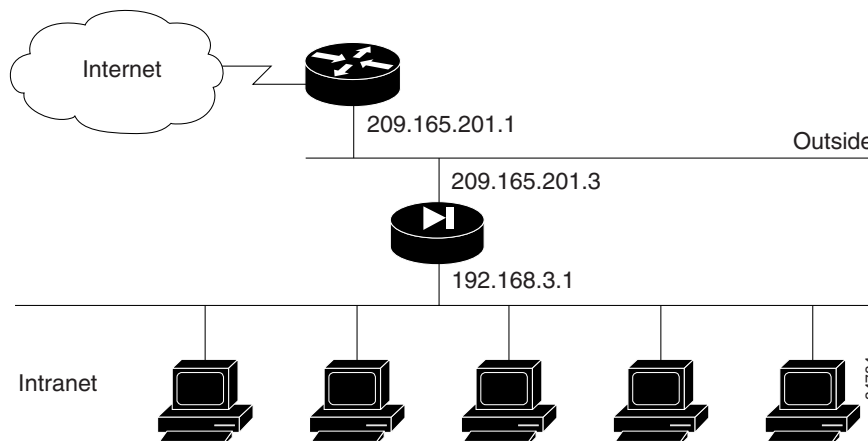
timeout uauth 0:05:00 absolute
no snmp-server location
no snmp-server contact
snmp-server community public
mtu outside 1500
mtu inside 1500

```

## 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 or PAT



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.

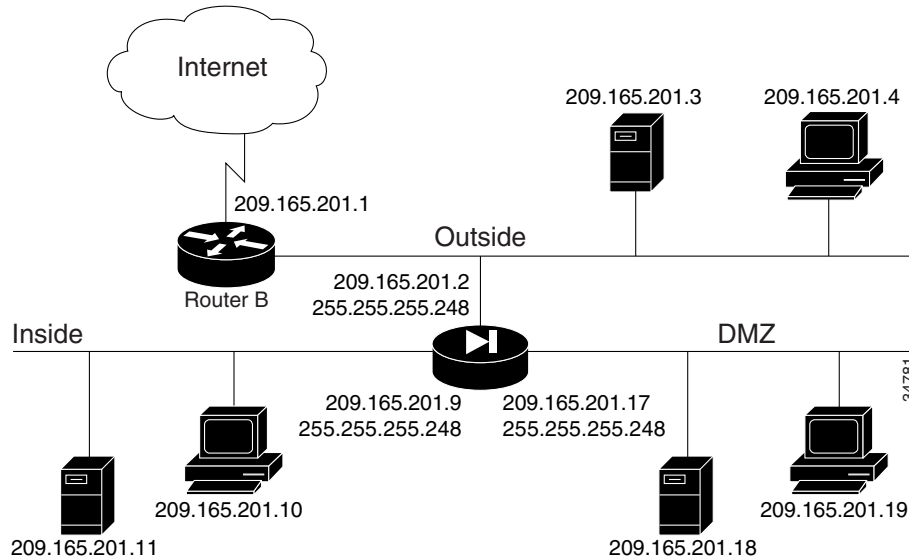
**Example 2-2 Two Interfaces with NAT**

```
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
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
```

## 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 without NAT or PAT**



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 ethernet0 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 209.165.201.2 255.255.255.248
ip address inside 209.165.201.9 255.255.255.248
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.

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

```
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 dmz security50
interface ethernet0 10baset
interface ethernet1 10baset
interface ethernet0 100basex
ip address outside 209.165.201.2 255.255.255.248
ip address inside 209.165.201.9 255.255.255.248
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 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
```

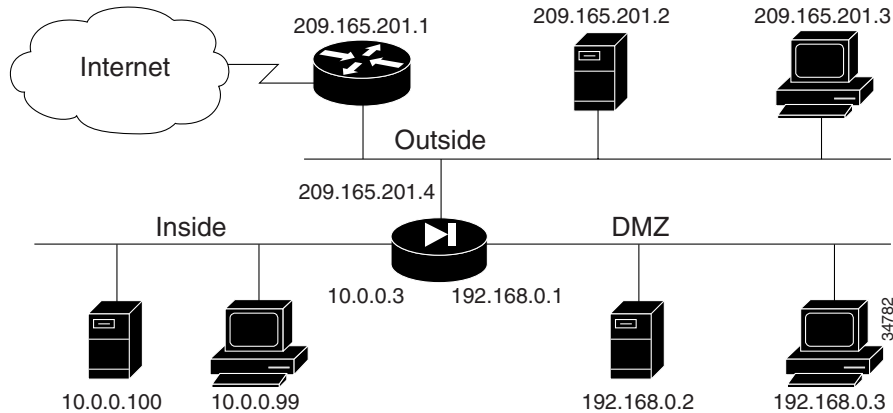
## Three Interfaces with NAT and PAT

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 “webserver.”

- 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 and PAT**



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 webservers
```

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

```
static (dmz,outside) 209.165.201.6 webservers netmask 255.255.255.255
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 and PAT**

```
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 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
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 netmask 255.255.255.255
access-list acl_out permit tcp any host 209.165.201.6 eq 80
access-group acl_out in interface outside
```

# Using Outside NAT

Starting with PIX Firewall version 6.2, NAT and PAT can be applied to traffic from an outside or less secure interface to an inside (more secure) interface. This functionality is called Outside NAT and provides the following benefits:

- Provides transparent support for Domain Name System (DNS)
- Simplifies routing by specifying the IP addresses that appear on the more secure interfaces of the PIX Firewall
- Enables connectivity between networks with overlapping IP addresses

For information about how Outside NAT enhances support for DNS, refer to the “[Basic Internet Protocols](#)” section in [Chapter 4, “Configuring Application Inspection \(Fixup\).”](#)



## Note

---

Outside NAT does not work with application inspection (“fixup”) for Internet Locator Service (ILS).

---

This section describes the last two scenarios and includes the following topics:

- [Overview](#)
- [Simplifying Routing](#)
- [Configuring Overlapping Networks](#)

## Overview

Outside NAT/PAT is similar to inside NAT/PAT, only the address translation is applied to addresses of hosts residing on the outer (less secure) interfaces of the PIX Firewall. To configure dynamic Outside NAT, specify the addresses to be translated on the less secure interface and specify the global address or addresses on the inside (more secure) interface. To configure static Outside NAT, use the **static** command to specify the one-to-one mapping.

After you configure Outside NAT, when a packet arrives at the outer (less secure) interface of the PIX Firewall, the PIX Firewall attempts to locate an existing xlate (address translation entry) in the connections database. If no xlate exists, it searches the NAT policy from the running configuration. If a NAT policy is located, an xlate is created and inserted into the database. The PIX Firewall then rewrites the source address to the mapped or global address and transmits the packet on the inside interface. Once the xlate is established, the addresses of any subsequent packets can be quickly translated by consulting the entries in the connections database.

To enable outside NAT, enter the following command:

```
nat interface natid access-list acl-name outside
```

Replace *interface* with the name of the lower security interface and replace *natid* with the identifier of the NAT entry. Replace *acl-name* with the name of any access list you want to apply. The **outside** option causes the translation of host addresses on the lower security interface. By default, address translation occurs only for host addresses on the higher security or “inside” interface.



## Note

---

If outside dynamic NAT is enabled on an interface, explicit NAT policy must be configured for all hosts on the interface.

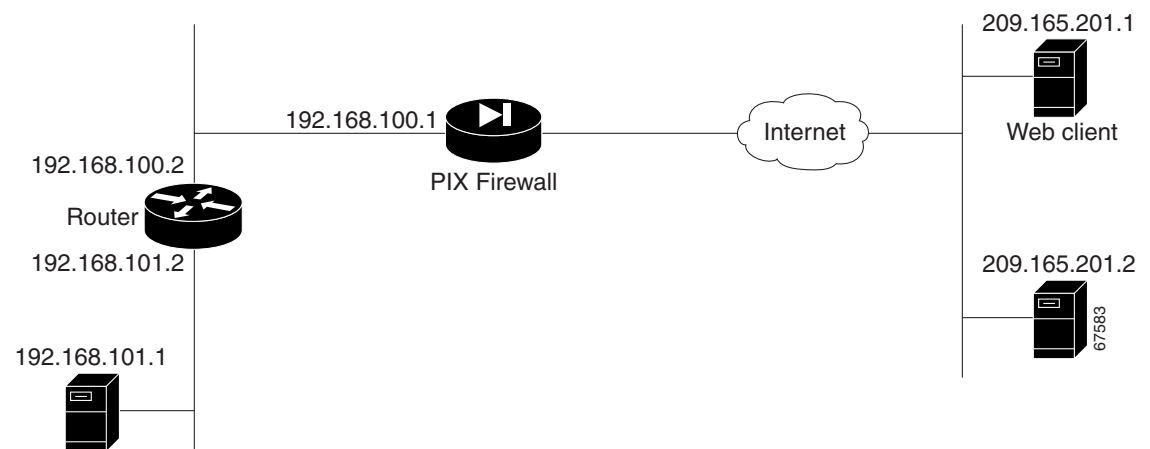
---

Use a *natid* of **0** with the **outside** option to disable address translation of hosts residing on the lower security interface. Use this option only if outside dynamic NAT is configured on the interface. By default, address translation is automatically disabled for hosts connected to the lower security interface.

## Simplifying Routing

You can use Outside NAT to simplify router configuration on your internal or perimeter networks by controlling the addresses that appear on these networks. For instance, in [Figure 2-8](#), the security policy allows clients in the network 209.165.201.0 to access only the servers on the internal network 192.168.101.0, including the web server 192.168.101.2.

**Figure 2-8** Simplifying Routing with Outside NAT



This policy can be supported by using the following command statements:

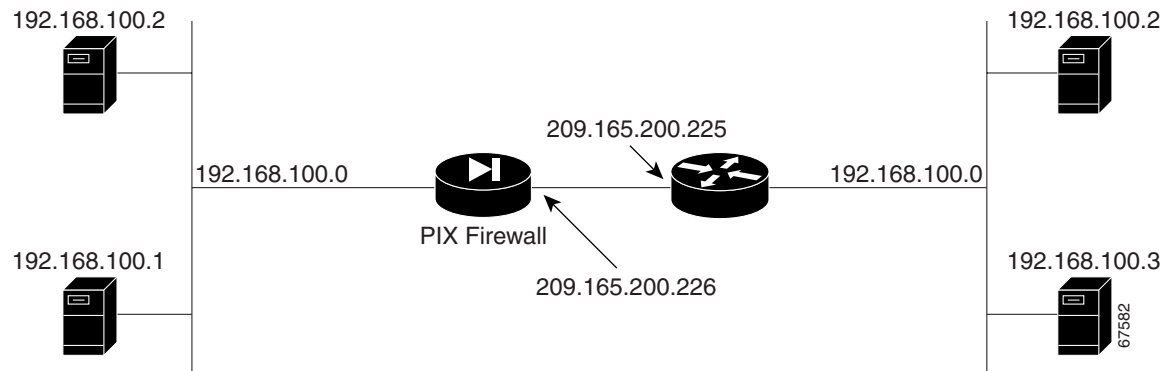
```
nat (outside) 1 209.165.201.0 255.255.255.0 outside
global (inside) 1 192.168.100.3-192.168.100.128
```

These commands translate all the source addresses on the remote network to a range of internal IP addresses (192.168.100.3-128). All other traffic for the network 192.168.101.0 arriving at the outside interface of the PIX Firewall is dropped. The router then automatically distributes the traffic from the inside interface of the PIX Firewall along with traffic originating on the 192.168.100.0 subnetwork.

## Configuring Overlapping Networks

In Figure 2-9, the PIX Firewall connects two private networks with overlapping address ranges.

**Figure 2-9 Using Outside NAT with Overlapping Networks**



In this example, an inside connected network, 192.168.100.0 255.255.255.0, overlaps with an outside network. In addition, a router (209.165.200.2) connects the outside interface of the PIX Firewall (209.165.200.3) to the outside network.

To enable connectivity between the two overlapping networks, the **alias** command can be used with previous versions of PIX Firewall, or static outside NAT can be used with PIX Firewall version 6.2 or later. We recommend using static outside NAT instead of the **alias** command because it allows the isolation of address translation between two interfaces and optionally supports rewriting of DNS address resource records.

The NAT command for translating the outside hosts from 192.168.100.0/24 into 209.165.200.0/24 on the inside network is as follows:

```
static (outside, inside) 209.165.200.0 192.168.100.0 netmask 255.255.255.0
```

The NAT command for translating the inside hosts from 192.168.100.0/24 into 209.165.200.0/24 on the outside network is as follows:

```
static (inside,outside) 209.165.200.0 192.168.100.0 netmask 255.255.255.0
```

In addition, the following routes need to be added in the PIX Firewall:

```
route inside 192.168.100.128 255.255.255.128 192.168.100.3 1
route inside 192.168.100.0 255.255.255.128 192.168.100.3 1
route outside 192.168.100.128 255.255.255.128 209.165.200.2 2
route outside 192.168.100.0 255.255.255.128 209.165.200.2 2
```



### Note

Splitting the netmask is required because an overlapping route cannot exist with a connected route.

With this configuration, an inside host 192.168.100.1 connects to address 209.165.200.3 to communicate with outside host 192.168.100.2.

# Enabling Stub Multicast Routing

This section describes how to implement the Stub Multicast Routing (SMR) feature, introduced with PIX Firewall version 6.2. It includes the following topics:

- [Overview](#)
- [Allowing Hosts to Receive Multicast Transmissions](#)
- [Forwarding Multicasts from a Transmission Source](#)
- [Configuring IGMP Timers](#)
- [Clearing IGMP Configuration](#)
- [Viewing and Debugging SMR](#)
- [For More Information about Multicast Routing](#)

## Overview

SMR allows the PIX Firewall to function as a “stub router.” A stub router is a device that acts as an Internet Group Management Protocol (IGMP) proxy agent. The IGMP is used to dynamically register specific hosts in a multicast group on a particular LAN with a multicast (MC) router. MC routers route multicast data transmissions to the hosts on each LAN in an internetwork that are registered to receive specific multimedia or other broadcasts. A stub router forwards IGMP messages between hosts and MC routers.

The Protocol Independent Multicast (PIM) protocol provides a scalable method for determining the best paths in a network for distributing a specific multicast transmission to each host that has registered using IGMP to receive the transmission. With PIM sparse mode (PIM SM), which is the default for Cisco routers, when the source of a multicast transmission begins broadcasting, the traffic is forwarded from one MC router to the next until the packets reach every registered host. If a more direct path to the traffic source exists, the last-hop router sends a join message toward the source that causes the traffic to be rerouted along the better path.

## Allowing Hosts to Receive Multicast Transmissions

When hosts that need to receive a multicast transmission are separated from the MC router by a PIX Firewall, configure the PIX Firewall to forward IGMP reports from the downstream hosts and to forward multicast transmissions from the upstream router. The upstream router is the next-hop interface toward the transmission source from the outside interface of the PIX Firewall.

To allow hosts to receive multicast transmissions through the PIX Firewall, perform the following steps:

---

**Step 1** Enable multicast forwarding on each interface by entering the following command:

```
multicast interface interface-name
```

This command enables multicast support on the specified interface and places the interface in multicast promiscuous mode. When you enter this command, the CLI enters multicast subcommand mode and the prompt changes to identify the interface you are configuring.

To use this command, replace *interface-name* with the name of the PIX Firewall interface on which you wish to enable multicast forwarding.

- Step 2** Configure the maximum number of IGMP groups, by entering the following command from multicast subcommand mode:

```
igmp max-groups n
```

To use this command, replace *n* with the maximum number of IGMP groups you wish to allow on the specified interface. The range of groups supported (max-groups) is from 1 to 2000. A value of 0 causes no IGMP groups to be allowed.

- Step 3** Enable IGMP forwarding on each PIX Firewall interface connected to hosts that will receive multicast transmissions. To do this, enter the following command from multicast subcommand mode for the interface, which is typically an inside (or more secure) interface.

```
igmp forward interface out-if-name
```

This command enables forwarding of all IGMP host reports and leaves messages received on the interface specified. To use this command, replace *out-if-name* with the name of the PIX Firewall interface that is connected to the MC router. This is typically the outside interface.

- Step 4** (Optional) Define static IGMP entries by using the following command:

```
[no] igmp join-group group-address
```

Enter this command on the downstream interface, which has receiving hosts in the multicast group.

This command configures the interface to be a statically connected member of the specified group. This allows the PIX Firewall to act for a client that may not be able to respond via IGMP, but still requires reception. This command is applied to the downstream interface towards the receiving hosts.

- Step 5** (Optional) Configure the multicast groups that hosts can join:

```
access-list acl_ID permit igmp any destination_addr destination_mask
```

This command configures an access control list that allows IGMP traffic to permissible Class D destination addresses.

- Replace *acl\_ID* with the name of the access control list.
- Replace *destination\_addr* with the Class D address of the multicast group from which you wish to allow hosts to receive multicast transmissions. To define many multicast groups with a single command, use the object grouping feature, described in “[Simplifying Access Control with Object Grouping](#)” in Chapter 3, “[Controlling Network Access and Use](#).”

- Step 6** Apply the access list by entering the following command from the multicast subcommand mode:

```
igmp access-group acl_ID
```

This command applies the access list to the currently interface.

### Example 2-5 Inside Receiving Hosts

In the following example, inside clients need to register with the multicast group with the Class D address 224.1.1.1:

```
multicast interface outside
igmp join-group 224.1.1.1 output-interface inside
```

After entering these commands, the PIX Firewall will act as an interested host for 224.1.1.1 and act accordingly on the interface to which the command was applied. Other downstream interfaces may be added to the list dynamically via IGMP.

**Example 2-6 Inside Receiving Hosts with Access Control**

The following example configures the inside and DMZ receivers:

```
multicast interface inside
igmp forward interface outside
igmp access-group 1
multicast interface dmz
    igmp forward interface outside
    igmp access-group 1
multicast interface outside
    igmp access-group 1

access-list 1 permit ip any 224.2.2.0 255.255.255.248
access-list 1 permit ip any 239.2.2.0 255.255.255.248
access-list 1 deny ip any any
```

---

## Forwarding Multicasts from a Transmission Source

When a multicast transmission source is on the inside (or more secure) interface of a PIX Firewall, you must configure the PIX Firewall to enable multicast forwarding from the source. You enable multicast forwarding on the PIX Firewall interfaces towards each network containing hosts that are registered to receive multicast transmissions from the source.

To configure the PIX Firewall to forward multicast transmissions from a source, perform the following steps:

---

**Step 1** Enable multicast forwarding on each PIX Firewall interface by entering the following command:

```
multicast interface interface-name
```

This command enables multicast support on the specified interface and places the interface in multicast promiscuous mode. When you enter this command, the CLI enters multicast subcommand mode and the prompt changes to identify the interface you are configuring.

To use this command:

- Replace *interface-name* with the name of the PIX Firewall interface on which you wish to enable multicast forwarding.

**Step 2** Create a static route from the transmission source to the next-hop router interface:

```
[no] mroute src smask in-if-name dst dmask out-if-name
```

- Replace *src* and *smask* with the IP address and subnet mask of the multicast source.
  - Replace *in-if-name* with the name of the PIX Firewall interface connected to the multicast source. This is typically the inside (or more secure) interface.
  - Replace *dst* and *dmask* with the Class D address and subnet mask for the multicast transmission from the source.
  - Replace *out-if-name* with the name of the PIX Firewall interface connected to the next-hop router interface toward the hosts registered to receive the transmission. This is typically the outside (or less secure) interface.
-

**Example 2-7 Inside Transmission Sources**

The following example configures the inside and DMZ sources with no internal receivers:

```
multicast interface outside
multicast interface inside
multicast interface dmz
mroute 1.1.1.1 255.255.255.255 inside 230.1.1.2 255.255.255.255 outside
mroute 2.2.2.2 255.255.255.255 dmz 230.1.1.2 255.255.255.255 outside
```

## Configuring IGMP Timers

This section describes how to change the default values for IGMP timers and includes the following topics:

- [Setting the Query Interval](#)
- [Setting Query Response Time:](#)

### Setting the Query Interval

Use the following command to configure the frequency at which IGMP query messages are sent by the interface.

```
[no] igmp query-interval seconds
```

The default is 60 seconds. To set the query interval back to the default, use the **no igmp query-interval** command.

### Setting Query Response Time

Use the following command to change the maximum query response time (for IGMP version 2 only).

```
[no] igmp query-max-response-time seconds
```

The default is 10 seconds. To set the query response time back to the default, use the **no igmp query-max-response-time** command.

## Clearing IGMP Configuration

This section describes how to clear IGMP entries.

Use the following command to delete entries from the IGMP cache:

```
clear igmp group [group-addr | interface interface-name]
```

Replace *group-addr* with the multicast group IP address. Replace *interface-name* with the interface name on your PIX Firewall on which IGMP is enabled.

Use the following command to clear static multicast routes:

```
clear mroute [src-addr | group-addr | interface interface_name]
```

Replace *src-addr* with the IP address of the multicast source. Replace *group-addr* with the address of the receiving multicast group. Replace *interface-name* with the PIX Firewall interface on which multicasts are enabled.

## Viewing and Debugging SMR

This section describes commands that you can use to view the current Multicast and IGMP configuration and for enabling debugging.

To display all or per-interface multicast settings, enter the following command:

```
show multicast [interface interface-name]
```

This also displays IGMP configuration for the interface. To use this command, replace *interface-name* with the name of the interface for which you wish to view configuration settings.

To display multicast-related information about one or more groups, enter the following command:

```
show igmp groups [group-address | interface interface-name]
```

Replace *group-address* with the Class D IP address of the group and replace *interface-name* with the name of the interface connected to the network where the groups are registered. To show all static multicast routes, enter the following command:

```
show mroute [src-address | group-address | interface interface_name]
```

Replace *src-address* with the IP address of the multicast transmission source or replace *group-address* with the Class D IP address of the group. Replace *interface-name* with the name of the interface connected to the network where the groups are registered.

To enable (or disable) debugging for IGMP events, enter the following command:

```
[no] debug igmp
```

To enable (or disable) debugging for multicast forwarding events, enter the following command:

```
[no] debug mfwrd
```

## For More Information about Multicast Routing

The following Cisco public websites provide background information about multicast routing:

[http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/ipimt\\_ov.htm](http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/ipimt_ov.htm)

<http://www.cisco.com/warp/public/732/Tech/multicast/>

The following RFCs from the IETF provide technical details about the IGMP and multicast routing standards used for implementing the SMR feature:

- RFC 2236 IGMPv2
- RFC 2362 PIM-SM
- RFC 2588 IP Multicast and Firewalls
- RFC 2113 IP Router Alert Option
- IETF draft-ietf-idmr-igmp-proxy-01.txt

