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:

Initial Configuration Checklist

Table 2-1 summarizes the tasks you should perform when you first configure your PIX Firewall to establish unrestricted outbound connectivity through the firewall. For instructions for controlling outbound connectivity or establishing inbound connectivity, see Chapter 3, "Controlling Network Access and Use."

Table 2-1 Initial Configuration Checklist

Task	Explanation	Procedure
If you have purchased a new feature license, upgrade your feature license	If you have purchased (or need to purchase) a new activation key for your PIX Firewall, upgrade your license before configuring the firewall.	Refer to the "Upgrading Your License by Entering a New Activation Key" section on page 11-2 in Chapter 11, "Changing Feature Licenses and System Software."
Deny ICMP traffic to the outside interface	By default, the PIX Firewall denies all inbound traffic through the outside interface. Before enabling inbound connectivity through the outside interface, you should consider configuring the PIX Firewall to deny all ICMP traffic to the outside interface. If no ICMP control list is configured, then the PIX Firewall accepts all ICMP traffic that terminates at any interface, including the outside interface.	To deny all ICMP traffic, including ping requests, through the outside interface, enter the following command: `icmp deny any outside` Enter this command for each additional interface on which you want to deny ICMP traffic. **Note** To test connectivity through the outside interface, temporarily change this setting, as described in the "Testing and Saving Your Configuration" section on page 2-22. For more information about the **icmp** command, refer to the *Cisco PIX Firewall Command Reference.*

Prevent fragmented packets	By default, the PIX Firewall accepts up to 24 fragments to reconstruct a full IP packet. Based on your network security policy, you should consider configuring the PIX Firewall to prevent fragmented packets from traversing the firewall.
	The PIX Firewall FragGuard feature provides IP fragmentation protection even without explicitly denying fragmented packets.
		To prevent fragmented packets on the outside and inside interfaces enter the following command:
		`fragment chain 1 outside`
		`fragment chain 1 inside`
		Enter this command for each additional interface on which you want to prevent fragmented packets.
		Note Adjust this setting to allow Network File System (NFS) connectivity through the interface.
		Setting the limit to 1 means that all packets must be unfragmented.
		For more information about the **fragment** command, refer to the *Cisco PIX Firewall Command Reference.*

Set default routes	Configure the default routes on your routers and hosts to forward traffic to the PIX Firewall.	Refer to the "Setting Default Routes" section on page 2-3.

Table 2-1 Initial Configuration Checklist (continued)

Task	Explanation	Procedure
Configure PIX Firewall interfaces	Assign an IP address and subnet mask to each interface in your PIX Firewall that connects to another network. All interfaces in a new PIX Firewall are shut down by default. You need to explicitly enable each interface you are using. Security levels let you control access between systems on different interfaces. You can use the default interface names and security levels or change them according to your security policy.	Refer to the "Configuring PIX Firewall Interfaces" section on page 2-4.
Configure the PIX Firewall for routing	You can configure each inside or perimeter PIX Firewall interface for the Routing Information Protocol (RIP) or Open Shortest Path First (OSPF) routing protocol. You can also configure the PIX Firewall to broadcast an inside or perimeter interface as a "default" route.	Refer to the "Configuring the PIX Firewall for Routing" section on page 2-13.
Establish outbound connectivity	Enable 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.	Refer to the "Testing and Saving Your Configuration" section on page 2-22.
Test connectivity	Temporarily enable ICMP messages to test that a host is reachable through the PIX Firewall.	Refer to the "Testing and Saving Your Configuration" section on page 2-22.
Save your configuration	When you complete entering commands in the configuration, save it to Flash memory and then reboot the PIX Firewall.	Refer to the "Saving Your Configuration" section on page 2-25.

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

A route, which is either statically defined or dynamically discovered, specifies the path used by a router or host to forward IP packets. You must define a special route, called the default route, for forwarding packets when no route is known. Packets destined for an unknown network are forwarded to the default router, which is sometimes called the gateway of last resort.

To configure the default routes on a Cisco IOS router to forward traffic to the PIX Firewall complete 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 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 if_address
```

For each PIX Firewall interface that is connected to a router, replace *if_address* with the IP address of the PIX Firewall interface.

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 **Cntrl-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 PIX Firewall interface and 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. Refer to the documentation for the operating system of a specific host for instructions for setting the default route.

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

Assign an IP address to each interface in your PIX Firewall that connects to another network. PIX Firewall interfaces do not have IP addresses until you assign them.

Note Multiple IP addresses can be assigned on the outside interface for internal web servers.

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**. Use the **nameif** command to change the default name of an interface.
- 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 appropriate network mask for the IP subnetwork.

  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 (from those that begin from 192 to 223). 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 address, 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 for the inside interface. Then enter the IP **address** command for the perimeter interface, including the 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. For details about the interface numbering of a specific PIX Firewall model, refer to the *Cisco PIX Firewall Hardware Installation Guide*.
- Replace *hardware_speed* with the speed of the interface, using the values shown in Table 2-2.

Note We recommend that you use the **auto** option to allow the PIX Firewall to automatically select the correct speed and duplex setting. If you use a fixed setting and you later change the setting, the interface will shut down.

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
10baset	10 Mbps Ethernet half-duplex communications.
100basetx	100 Mbps Ethernet half-duplex communications.
100full	100 Mbps Ethernet full-duplex communications.
1000full	1000 Mbps Gigabit Ethernet, autonegotiaties advertising full duplex only.
1000full nonegotiate	1000 Mbps Gigabit Ethernet, forces speed to 1000 Mbps full duplex.
1000auto	1000 Mbps Gigabit Ethernet to auto-negotiate full or half duplex.
aui	10 Mbps Ethernet half-duplex communications for an AUI cable interface.
auto	Automatically sets Ethernet speed and duplex operation.
bnc	10 Mbps Ethernet half-duplex communications for a BNC cable interface.

Note Make sure the maximum transmission unit (MTU) is no more than 1500 bytes for Ethernet. To view the MTU, use the **show mtu** command.

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 security100. 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 **ethernet***0*.
- 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 intf*n*, 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 = security20, and intf5 = security25. You can choose any unique security level between 1 and 99 for a perimeter interface.

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

Network Address Translation (NAT) allows you to hide internal IP addresses, those behind the PIX Firewall, from external networks. NAT is accomplished by mapping global IP addresses to local IP addresses. Static NAT is described in the "Enabling 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 addresses has up to 65,535 ports that are available for making connections. For PAT, the port number uniquely identifies each connection.

Usually, 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 and higher supports NAT and PAT of addresses on outside networks (lower security interfaces) that initiate connections to hosts on higher 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 the "Using Outside NAT" section on page 2-38."

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 addresses 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 external addresses. 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 "How Application Inspection Works" section on page 5-1 in Chapter 5, "Configuring Application Inspection (Fixup)."
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 a host address on a less secure interface and a range or pool of IP addresses on a more secure interface. This provides a one-to-one mapping between an external and an internal address. 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.

Table 2-3 Address Translation Types

Type of Address Translation	Function
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.
Policy NAT	Translates source and destination address pairs to different global statements, even if the source address is the same. For example, traffic from IP address A to server A can be translated to global address A, while traffic from IP address A to server B can be translated to global address B.

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 ID 5, 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 ID 5.

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.

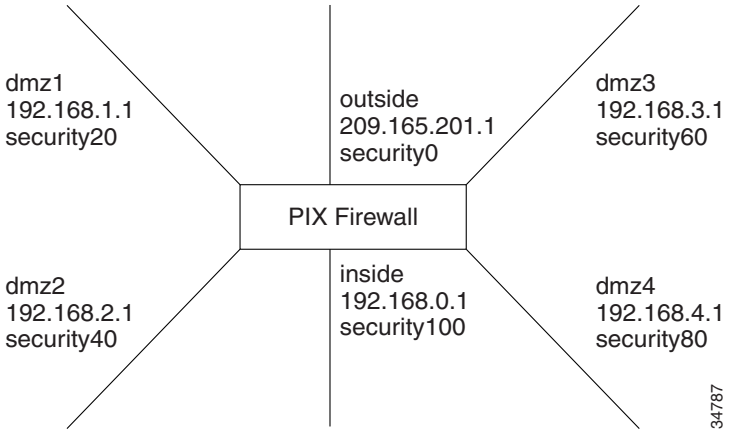
Configuring NAT and PAT

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 2-1.

Figure 2-1 Sketching Interfaces and Security Levels



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

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

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

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

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

e. To let **dmz1** users start connections to the outside, use the **nat (dmz1) 1 0 0** command.

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

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

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

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

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

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

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

There should be enough global addresses to handle the number of users on each interface simultaneously accessing 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 configures a pool of global addresses on the outside interface.

When you define IP address pools for NAT and PAT in the same configuration for the same interface, the PIX Firewall uses the NAT address pools first, regardless of the order of the statements in the configuration. If there is more than one statement assigning IP address pools for NAT, the addresses are used in the order of the statements. IP addresses assigned for PAT are used only after any NAT IP address pools are exhausted. This minimizes the exposure of PAT in case 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.

Configuring the PIX Firewall for Routing

A route identifies the interface and router (gateway) to use to forward packets for a specific destination network received by the PIX Firewall. This section describes how to configure the PIX Firewall to correctly route traffic to and from adjacent networks. It includes the following topics:

Using RIP

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.

If the PIX Firewall has RIP enabled, it 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 when defining a default route.

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.

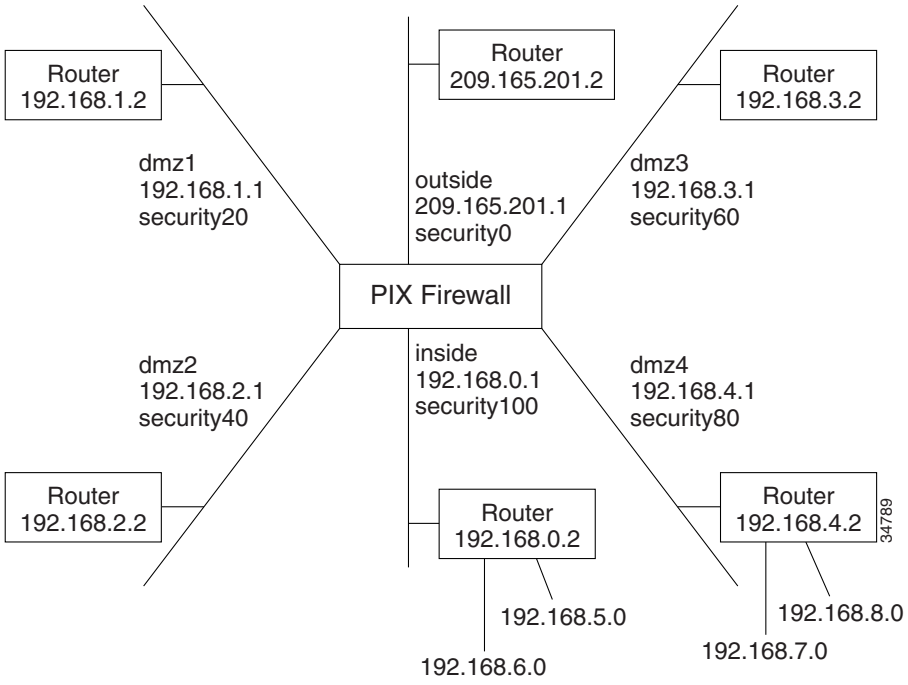
Note Before testing your configuration, flush the ARP caches on any routers that feed traffic into or from the PIX Firewall and between the PIX Firewall and the Internet. For Cisco routers, use the **clear arp** command to flush the ARP cache.

Configuring RIP 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 2-2.

Figure 2-2 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.

Using OSPF

This section describes how the Open Shortest Path First (OSPF) routing protocols are implemented in PIX Firewall Version 6.3. It includes the following topics:

Overview

PIX Firewall Version 6.3 introduces support for dynamic routing using the Open Shortest Path First (OSPF) routing protocol. OSPF is widely deployed in large internetworks because of its efficient use of network bandwidth and its rapid convergence after changes in topology.

Note OSPF is not supported on the PIX Firewall 501.

The OSPF functionality in PIX Firewall Version 6.3 is similar to that provided by Cisco IOS Release12.2(3a). For details about the syntax for each command and subcommand used to manage OSPF, refer to the *Cisco PIX Firewall Command Reference* or to Cisco IOS software documentation.

Security Issues When Using OSPF

Authentication should be used with all routing protocols when possible because route redistribution between OSPF and other protocols (like RIP) can potentially be used by attackers to subvert routing information. If MD5 authentication is used on all segments, security should not be an issue with OSPF.

When using dynamic routing, the physical security of the PIX Firewall is of increased importance. Access to the physical device and configuration information should be kept secure. Shared-keys should be changed at a reasonable interval.

As part of its normal operation, OSPF advertises routes to networks, and this may not be desirable in many PIX Firewall implementations. You may need to prevent networks from being advertised externally when using private addressing or when required by your security policy.

If NAT is used, if OSPF is operating on public and private areas, and if address filtering is required, then you need to run two OSPF processes—one process for the public areas and one for the private areas.

A router that has interfaces in multiple areas is called an Area Border Router (ABR). A router that redistributes traffic or imports external routes (Type 1 or Type 2) between routing domains is called an Autonomous System Boundary Router (ASBR).

An ABR uses link-state advertisements (LSA) to send information about available routes to other OSPF routers. Using ABR type 3 LSA filtering, you can have separate private and public areas with the PIX Firewall acting as an ABR. Type 3 LSAs (inter-area routes) can be filtered from one area to another. This lets you use NAT and OSPF together without advertising private networks.

Note Only type 3 LSAs can be filtered. If you configure PIX Firewall as an ASBR in a private network, it will send type 5 LSAs describing private networks, which will get flooded to the entire AS including public areas.

If NAT is employed but OSPF is only running in public areas, then routes to public networks can be redistributed inside the private network, either as default or type 5 AS External LSAs. However, you need to configure static routes for the private networks protected by the PIX Firewall. Also, you cannot mix public and private networks on the same PIX Firewall interface.

OSPF Features Supported

The following is a list of OSPF features supported by PIX Firewall Version 6.3:

- Support of intra-area, inter-area and External (Type I and Type II) routes
- Support for virtual links
- OSPF LSA flooding
- Authentication for OSPF packets (both clear text and MD5 authentication)
- Support for configuring the PIX Firewall as a designated router (DR) or ABR
- Support for configuring the PIX Firewall as an ASBR, with route redistribution between OSPF processes including OSPF, static, and connected routes
- Support for stub areas and not so stubby areas (NSSA)
- ABR type 3 LSA filtering
- Load balancing among a maximum of three peers on a single interface, using Equal Cost Multipath Routes (ECMP).

Note If using ECMP, note that the default cost for a Fast Ethernet link on the PIX Firewall is consistent with a Cisco Firewall Services Module (FWSM) but differs from a Cisco IOS router.

Table 2-4 summarizes the OSPF commands that are supported or that are not supported in PIX Firewall Version 6.3. For the detailed syntax of each command, refer to the Cisco IOS Release 12.2(3a) documentation or to the *Cisco* PIX Firewall *Command Reference*.

Table 2-4 Cisco IOS OSPF Commands Supported in PIX Firewall Version 6.3

OSPF Command[1]	Supported	OSPF Command	Supported	OSPF Command	Supported
`area authentication`	yes	`ip ospf dead-interval`	yes	`show ip ospf flood-list`	yes
`area default-cost`	yes	`ip ospf flood-reduction`	no	`show ip ospf interface`	yes
`area filter-list`	yes	`ip ospf hello-interval`	yes	`show ip ospf neighbor`	yes

Table 2-4 Cisco IOS OSPF Commands Supported in PIX Firewall Version 6.3 (continued)

OSPF Command[1]	Supported	OSPF Command	Supported	OSPF Command	Supported
area nssa	yes	ip ospf message-digest-key	yes	show ip ospf request-list	yes
area range	yes	ip ospf mtu-ignore	yes	show ip ospf retransmission-list	yes
area stub	yes	ip ospf name-lookup	no	show ip ospf summary-address	yes
area virtual-link	yes	ip ospf priority	yes	show ip ospf virtual-links	yes
auto-cost	no (use ospf cost)	ip ospf retransmit-interval	yes	summary-address (OSPF)	yes
compatible rfc1583	yes	ip ospf transmit-delay	yes	timers lsa-group-pacing	yes
default-information originate (OSPF)	yes	log-adj-changes	yes	timers spf	yes
distance ospf	yes	network area	yes	clear ip ospf	modified
ignore lsa mospf	yes	router-id	yes	default-metric (OSPF)	no
ip ospf authentication	yes	router ospf	yes	ip ospf demand-circuit	no
ip ospf authentication-key	yes	show ip ospf [process-id]	yes	ip ospf network	no
ip ospf cost	yes	show ip ospf border-routers	yes	neighbor (OSPF)	no
ip ospf database-filter	yes	show ip ospf database	yes		

1. The exact syntax for some commands used with PIX Firewall may differ slightly from the Cisco IOS software implementation. Refer to the *Cisco PIX Firewall Command Reference* for the exact syntax of a specific command.

PIX Firewall does not accept spaces within OSPF authentication keys or message digests but Cisco IOS does. This may create compatibility issues when a PIX Firewall tries to exchange OSPF messages if an adjacent router uses spaces within its authentication key or message digest.

Restrictions and Limitations

The PIX Firewall does not provide any filtering of OSPF in Version 6.3 beyond what is provided by OSPF.

OSPF does not support dynamic routing over overlapping address spaces, so the PIX Firewall will not support running OSPF on an interface from where it can learn overlapping addresses. To support overlapping address networks, either configure static routes or use passive RIP.

Note Running both OSPF and RIP concurrently on the same PIX Firewall is unsupported.

Only broadcast networks are supported by the implementation of OSPF in PIX Firewall Version 6.3. The following summarizes the OSPF features that are *not* supported by PIX Firewall Version 6.3:

- Point-to-point link/serial interface/nonbroadcast multiaccess (NBMA)
- OSPF on demand Circuit
- Flood Reduction

- Redistribution of routes between non-OSPF routing protocols
- Policy Routing

A maximum of two OSPF processes are allowed and PIX Firewall will only allow redistribution between these OSPF processes.

Any topology in which the same router is connected to two different interfaces of the PIX Firewall is not supported.

Note When you configure OSPF on either IOS or the PIX Firewall using the **default-information originate** command with the **always** keyword and a route-map with match clauses, there must be a route to match in the routing table. If there is no match, then the route is not redistributed. If a system is configured with the **always** keyword, it will not install a default route from another system. Also, do not configure a default route with the IP address of a PIX Firewall interface as a gateway.

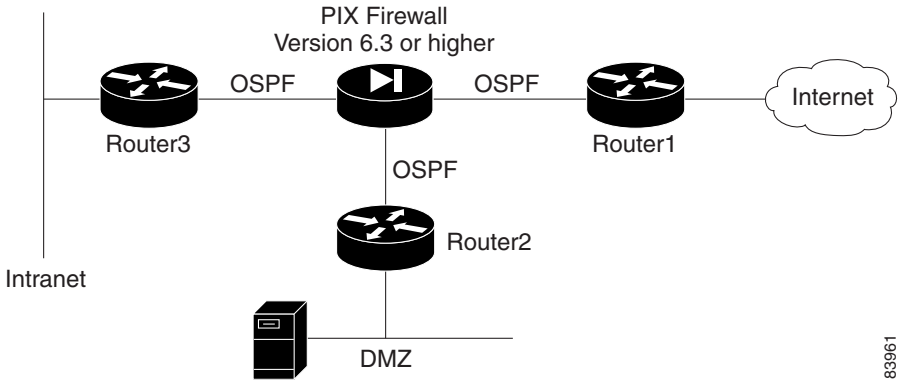
Configuring OSPF on the PIX Firewall

This section describes how to configure the PIX Firewall when using OSPF. It includes the following topics:

Using OSPF in Public Networks

Figure 2-3 illustrates an implementation of PIX Firewall using OSPF in public and private networks.

Figure 2-3 Using OSPF with PIX Firewall Version 6.3



This example illustrates the PIX Firewall as an ABR, configured to filter Type 3 LSAs, with NAT enabled on the inside interface, NAT disabled on the DMZ, and all interfaces running OSPF. Router1 is a locally controlled ASBR running OSPF and Border Gateway Protocol (BGP).

Note If NAT is enabled, but OSPF is running only in public areas, the only special configuration required is to configure static routes for the private networks protected by the PIX Firewall.

In this configuration, the inside interface learns routes dynamically from all areas, but its private routes are not propagated onto the backbone or public areas. The DMZ is visible to the backbone.

Follow these steps to configure this implementation on the PIX Firewall:

Step 1 To configure the PIX Firewall interfaces, enter the following commands:

```
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 dmz security50
ip address outside 1.1.1.1 255.255.255.0
ip address inside 10.0.0.1 255.0.0.0
ip address dmz 1.1.2.1 255.255.255.0
```

Step 2 To configure the static routes, enter the following commands:

```
static (inside,outside) 1.1.1.2 10.1.1.2 255.255.255.255
static (dmz,outside)1.1.2.2 1.1.2.2 255.255.255.255
```

Step 3 Configure NAT by entering the following commands:

```
nat (inside) 1 0 0
nat (dmz)0 0 0
global (outside) 1 1.1.1.4-1.1.1.254
```

Step 4 Configure OSPF by entering the following commands:

```
router ospf 1
area 0 filter-list prefix ten in
network 1.1.1.0 255.255.255.0 area 0
network 1.1.2.0 255.255.255.0 area 1.1.2.0
network 10.0.0.0 255.0.0.0 area 10.0.0.0
prefix-list ten deny 10.0.0.0/8
prefix-list ten permit 1.1.2.0/24
```

Example 2-1 Moving a Network to a Different OSPF Process

Before reassigning a network to a new OSPF process ID, remove the OSPF configuration line for the network that assigned it to the previous OSPF process ID. Then configure the new OSPF process ID assignment for that network.

The following example shows the configuration for an existing network:

```
router ospf 10
  distance ospf intra-area 130 inter-area 120
  log-adj-changes
router ospf 50
  network 10.130.12.0 255.255.255.0 area 10.130.12.0
  network 10.132.12.0 255.255.255.0 area 0
  network 10.139.12.0 255.255.255.0 area 50
  area 50 stub
  log-adj-changes
```

To move the network 10.130.12.0 255.255.255.0 area 10.130.12.0 to router ospf 10, enter the following commands:

```
pixfirewall(config-router)# router ospf 50
pixfirewall(config-router)# no network 10.130.12.0 255.255.255.0 area 10.130.12.0
pixfirewall(config-router)# router ospf 10
pixfirewall(config-router)# network 10.130.12.0 255.255.255.0 area 10.130.12.0
pixfirewall(config-router)# s router
router ospf 10
  network 10.130.12.0 255.255.255.0 area 10.130.12.0
  distance ospf intra-area 130 inter-area 120
  log-adj-changes
router ospf 50
  network 10.132.12.0 255.255.255.0 area 0
  network 10.139.12.0 255.255.255.0 area 50
  area 50 stub
  log-adj-changes
```

Using OSPF in Private and Public Networks

When NAT is used and OSPF is operating on public and private areas you need to run two OSPF processes to prevent the advertising of private networks in public areas. This lets you use NAT and OSPF, without advertising private networks.

In this implementation, the PIX Firewall is used as an ASBR with NAT enabled on both the inside interface and on the DMZ, with all interfaces running OSPF. This configuration allows both the inside and DMZ interfaces to learn routes dynamically from all areas, while preventing the private routes from being propagated onto the backbone or public areas.

Follow these steps to configure this implementation on the PIX Firewall:

Step 1 To configure the PIX Firewall interfaces, enter the following commands:

```
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 dmz security50
ip address outside 1.1.1.1 255.255.255.0
ip address inside 10.0.0.1 255.0.0.0
ip address dmz 192.168.1.1 255.255.255.0
```

Step 2 To configure the static routes, enter the following commands:

```
static (inside,outside) 1.1.1.2 10.1.1.2 255.255.255.255
static (dmz,outside)1.1.1.3 192.168.1.3 255.255.255.255
```

Step 3 Configure NAT by entering the following commands:

```
nat (inside) 1 0 0
nat (dmz)1 0 0
global (outside) 1 1.1.1.4-1.1.1.254
```

Step 4 Configure OSPF by entering the following commands:

```
router ospf 1 //public AS
network 1.1.1.0 255.255.255.0 area 0
router ospf 2 //private AS
redistribute ospf 1 //import the public external routes
network 10.0.0.0 255.0.0.0 area 10.0.0.0
network 192.168.1.0 255.255.255.0 area 192.168.1.0
```

Viewing OSPF Configuration

Table 2-5 lists some of the **show** commands that you can enter from privileged or configuration modes to display information about OSPF on the PIX Firewall. Refer to the *Cisco PIX Firewall Command Reference* or to the Cisco IOS documentation for all the options and the detailed syntax.

Table 2-5 OSPF show Commands

Command	Result
`show routing [interface interface-name]`	Displays non-default routing configuration information. Use the **interface** option to display information for a specific interface and replace *interface-name* with the identifier for a specific interface.
`show ospf [process-id]`	Displays general information about OSPF routing process IDs. Use the **process-ID** option to display information for a specific routing process.
`show ospf border-routers`	Displays the internal OSPF routing table entries to an Area Border Router (ABR) and Autonomous System Boundary Router (ASBR).
`show ospf database [router][network][external]`	Displays lists of information related to the OSPF database for a specific router. The different options display information about different OSPF link-state advertisements (LSAs).
`show ospf flood-list interface-name`	Displays a list of OSPF link-state advertisements (LSAs) waiting to be flooded over an interface. Replace *interface-name* with the identifier for a specific interface.
`show ospf interface interface-name`	Displays OSPF-related interface information. Use the **interface** option to display information for a specific interface and replace *interface-name* with the identifier for a specific interface.
`show ospf neighbor [interface-name] [neighbor-id] [detail]`	Displays OSPF neighbor information on a per-interface basis. Replace *interface-name* with the identifier for a specific interface. Use the **neighbor-id** option to display information about a specific neighbor. Use the **detail** option to display detailed information.
`show ospf request-list [neighbor-addr] [interface-name]`	Displays a list of all link-state advertisements (LSAs) requested by a router. Replace *neighbor-addr* with the IP address of a neighbor. Replace *interface-name* with the identifier for a specific interface.

Table 2-5 OSPF show Commands (continued)

Command	Result
`show ospf retransmission-list [neighbor-addr] [interface-name]`	Displays a list of all link-state advertisements (LSAs) waiting to be resent. Replace *neighbor-addr* with the IP address of a neighbor. Replace *interface-name* with the identifier for a specific interface.
`show ospf [process-id] summary-address`	Displays a list of all summary address redistribution information configured under an OSPF process. Replace process-ID with a process identifier for a specific OSPF area ID.
`show ospf virtual-links`	Displays parameters and the current state of OSPF virtual links.

Clearing OSPF Configuration

This section describes how to clear OSPF configuration.

To clear the OSPF routing process ID, use the following command:

```
clear ospf [pid] {process | counters neighbor [neighbor-intf] [neighbor-id]}
```

This command only clears the process ID and does not clear any configuration. Replace *pid* with the OSPF routing process ID. Replace *neighbor-intf* with the interface for a specific neighbor. Replace *neighbor-id* with the IP address of a specific neighbor.

To clear the OSPF configuration, use one of the following commands:

```
no routing interface interface-name>
no router ospf id
```

Replace *interface-name* with the identifier for a specific interface. Replace *id* with the OSPF area identifier.

Testing and Saving Your Configuration

This section describes how to make sure your configuration works by testing connectivity, and how to save your configuration. It includes the following topics:

Testing Connectivity

You can use the **access-list** command to allow hosts on one interface to ping through to hosts on another interface. This lets you test that a specific 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.

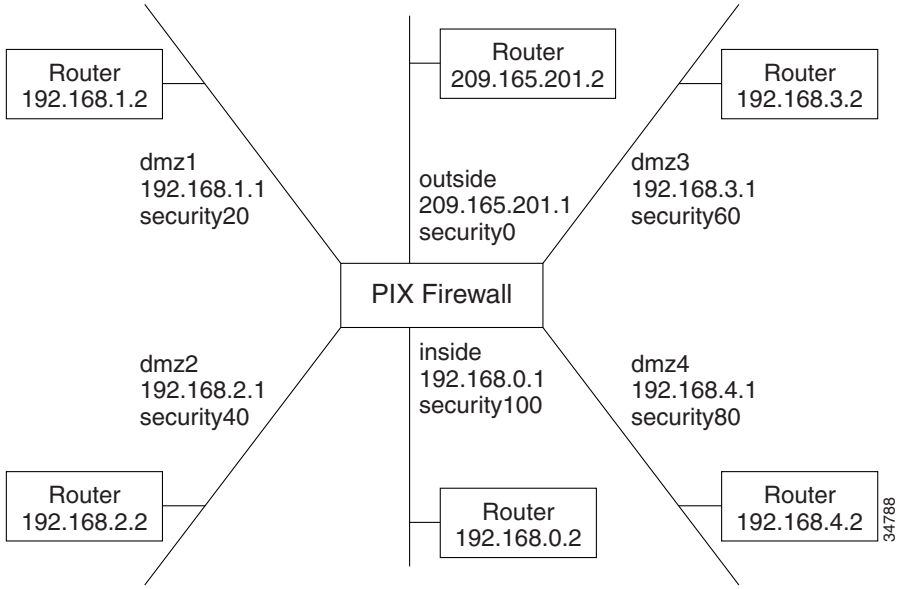
Note We recommend that you only enable pinging during troubleshooting. When you are done testing the interfaces, remove the ICMP **access-list** command statements.

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 2-4 shows an example sketch:

Figure 2-4 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 higher, 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, which shows 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. Verify the physical connectivity of the affected interface(s). If there are switches or hubs between the hosts and the PIX Firewall, verify that all the links are working. You can try connecting a host directly to the PIX Firewall using a crossover cable.

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

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

c. 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.

d. 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.

   If there is a single router between the host and the PIX Firewall, a default route on the router should be unnecessary. However, you might want to try clearing the ARP cache of the router. If there are multiple routers, you might need to set a default route on any router on the path from the PIX Firewall to the host.

e. 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.

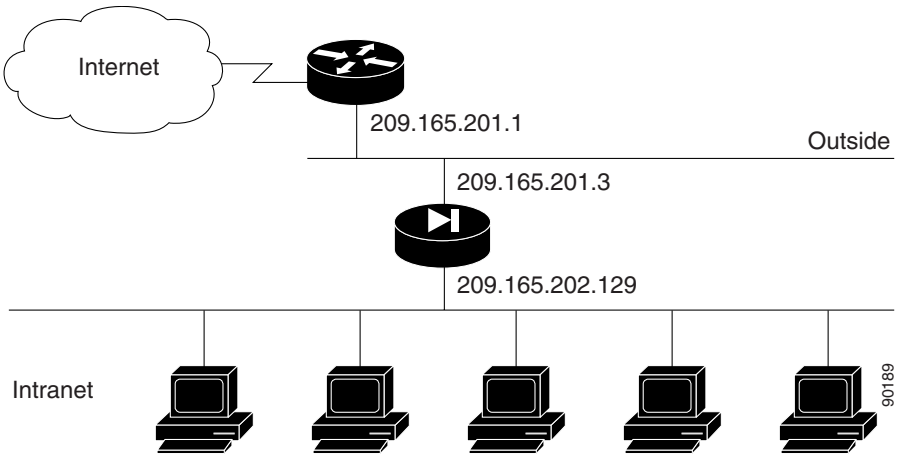
Basic Configuration Examples

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

When you first add a PIX Firewall to an existing network, it is easiest to implement if you do not have to renumber all the inside and outside IP addresses. The configuration in Figure 2-5 illustrates this scenario. All inside hosts can start connections. All external hosts are blocked from initiating connections or sessions on inside hosts.

Figure 2-5 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 100basetx
interface ethernet1 100basetx
```

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 209.165.202.129 255.255.255.0
```

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) 0 209.165.201.3 255.255.255.224
```

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 2-2 shows the listing for the basic configuration required to implement a PIX Firewall with two interfaces without NAT.

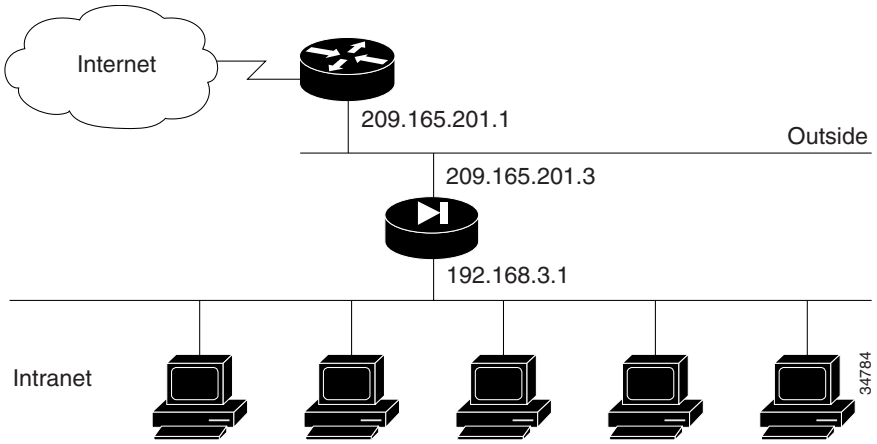
Example 2-2 Two Interfaces Without NAT

```
nameif ethernet0 outside security0
nameif ethernet1 inside security100
interface ethernet0 100basetx
interface ethernet1 100basetx
ip address outside 209.165.201.3 255.255.255.224
ip address inside 209.165.202.129 255.255.255.0
hostname pixfirewall
arp timeout 14400
no failover
names
pager lines 24
logging buffered debugging
nat (inside) 0 209.165.201.3 255.255.255.224
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 2-6 illustrates a network using unregistered IP addresses on the intranet, which requires NAT for connecting to the public Internet.

Figure 2-6 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 192.168.3.1 255.255.255.0
```

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 2-3 shows the complete configuration for configuring two interfaces with NAT.

Example 2-3 Two Interfaces with NAT

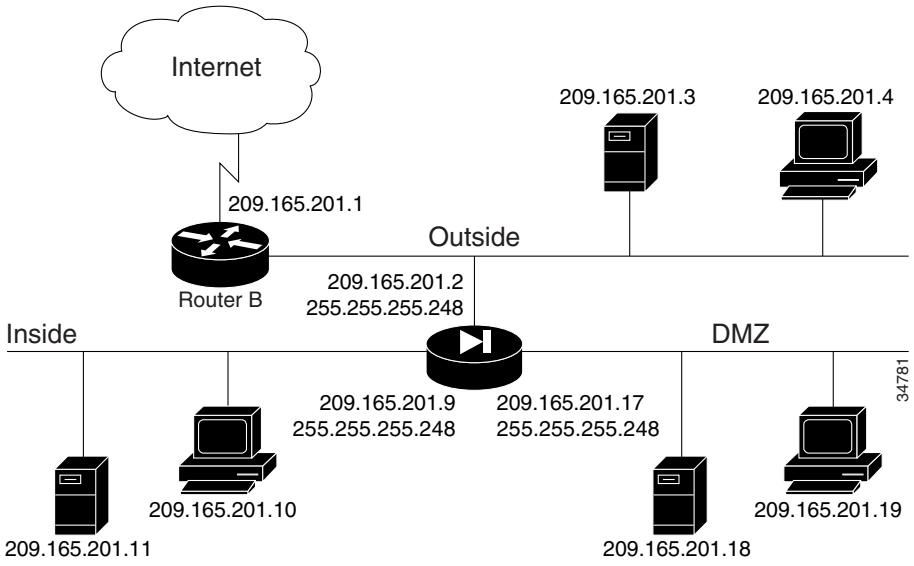
```
nameif ethernet0 outside security0
nameif ethernet1 inside security100
interface ethernet0 100basetx
interface ethernet1 100basetx
ip address outside 209.165.201.3 255.255.255.224
ip address inside  192.168.3.1 255.255.255.0
hostname pixfirewall
arp timeout 14400
```

```
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 2-7, the PIX Firewall has three interfaces configured without address translation.

Figure 2-7 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

In addition, the DMZ host 209.165.201.19 must be accessible from hosts on the outside interface.

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 100basetx
interface ethernet1 100basetx
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 209.165.201.2 255.255.255.248
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) 209.165.201.2 209.165.201.19 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 2-4 shows the complete configuration for three interfaces without NAT.

Example 2-4 Three Interfaces Without NAT or PAT

```
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 dmz security50
hostname pixfirewall
names
access-list acl_out permit tcp any host 209.165.201.19
access-list acl_out permit icmp any any
access-list ping_acl permit icmp any any
pager lines 24
logging buffered debugging
interface ethernet0 100basetx
interface ethernet1 100basetx
interface ethernet2 100basetx
mtu outside 1500
mtu inside 1500
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
```

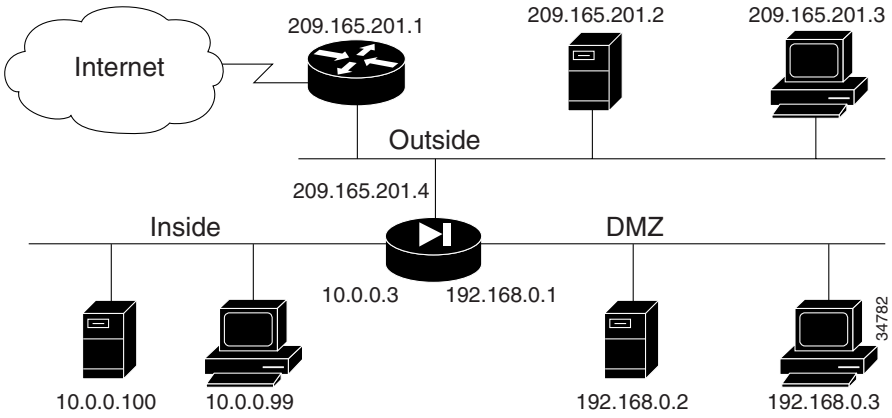
```
no failover
arp timeout 14400
nat (inside) 0 209.165.201.8 255.255.255.248
static (dmz,outside) 209.165.201.2 209.165.201.19 netmask 255.255.255.248
access-group acl_out in interface outside
access-group ping_acl in interface inside
access-group ping_acl in interface dmz
route outside 0.0.0.0 0.0.0.0 209.165.201.1 1
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
```

Three Interfaces with NAT and PAT

In Figure 2-8, 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 private 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 2-8 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 Enable Telnet access for a host on the inside interface of the PIX Firewall by entering the following commands:

```
telnet 10.0.0.100 255.255.255.255
telnet timeout 15
```

Step 2 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 3 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 4 Give the IP address of the web server a label:

```
name 192.168.0.2 webserver
```

Step 5 Let any user on the outside interface access the web server on the DMZ interface:

```
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
```

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

Example 2-5 Three Interfaces with NAT and PAT

```
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 dmz security50
hostname pixfirewall
names
name 192.168.0.2 webserver
access-list acl_out permit icmp any any
access-list acl_out permit tcp any host 209.165.201.6 eq 80
access-list ping_acl permit icmp any any
pager lines 24
logging buffered debugging
interface ethernet0 100basetx
interface ethernet1 100basetx
interface ethernet2 100basetx
mtu outside 1500
mtu inside 1500
mtu dmz 1500
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
no failover
arp timeout 14400
```

```
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
static (dmz,outside) 209.165.201.6 webserver netmask 255.255.255.255
access-group acl_out in interface outside
access-group ping_acl in interface inside
access-group ping_acl in interface dmz
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
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
telnet 10.0.0.100 255.255.255.255
telnet timeout 15
```

Using VLANs with the Firewall

PIX Firewall Version 6.3 introduces support for VLANs. This section describes how to use and implement VLANs with firewall and includes the following topics:

Overview

Virtual LANs (VLANs) are used to create separate broadcast domains within a single switched network. Some of the benefits of VLANs include the following:

- Broadcast control
- Improved security
- Flexibility
- Scalability

A VLAN can be created through software configuration whenever it is needed because no actual separation is required in the physical or data link network. To create a VLAN, you simply assign ports on each switch to the new VLAN. However, the VLAN must then be interconnected to the rest of your network through a router or other device that can forward packets between the ports assigned to the VLAN.

Note When configuring failover for a VLAN interface, hello packets are sent over the physical interface, so the physical interface must be configured with an ip address.

Using Logical Interfaces

With Version 6.3, you can assign VLANs to physical interfaces on the PIX Firewall, or you can configure multiple logical interfaces on a single physical interface and assign each logical interface to a specific VLAN.

A logical interface is similar in many respects to a so-called physical interface. Both logical and physical interfaces are software objects (the actual *physical* object is the network interface card on the PIX Firewall unit). What is called the physical interface for the purpose of configuration is a software object that has both Layer 2 (Data link) and Layer 3 (Network) attributes. Layer 2 attributes include maximum transmission unit (MTU) size and failover status, while Layer 3 attributes include IP address and security level.

A logical interface has only Layer 3 attributes. As a result, you can issue certain commands, such as **failover link** *if_name* or **failover lan interface** *if_name* on a physical interface that you cannot use with a logical interface. When you disable a physical interface, all the associated logical interfaces are also disabled. When you disable a logical interface, it only affects the logical interface.

Note Failover is supported with VLAN interfaces. But the **failover lan interface** command does not support VLAN interfaces or the **failover link** commands.

The number of logical interfaces that you can configure varies according to the model. The minimum number of interfaces for any PIX Firewall is two. Table 2-6 lists the maximum number of logical interfaces supported on a specific PIX Firewall model:

Table 2-6 Maximum Number of Interfaces Supported on PIX Firewall Models

Model	Restricted License[1]			Unrestricted License		
	Total Interfaces	Physical Interfaces	Logical Interfaces	Total Interfaces	Physical Interfaces	Logical Interfaces
PIX 501[2]	NA	NA	NA	2	2	Not supported
PIX 506/506E	NA	NA	NA	4	2	2
PIX 515/515E	5	3	3	10	6	8
PIX 520[3]	NA	NA	NA	12	6	10
PIX 525	8	6	6	12	8	10
PIX 535	10	8	8	24	10	22

1. PIX 501 and PIX 506/506E do not support Restricted/Unrestricted licenses.
2. One interface of the PIX 501 connects to an integrated 4-port switch.
3. PIX 520 supports a connection license and the number of interfaces does not vary with the connection license.

Note To determine the maximum number of logical interfaces that you can use, subtract the number of physical interfaces in use on your PIX Firewall from the number of total interfaces.

VLAN Security Issues

By default, with no VLANs configured, the PIX Firewall sends untagged packets to any directly connected switch. If an untagged packet is received by a switch on a trunk port, the switch forwards the packet on the native VLAN assigned for that trunk port. By default, switches assign VLAN 1 to the native VLAN.

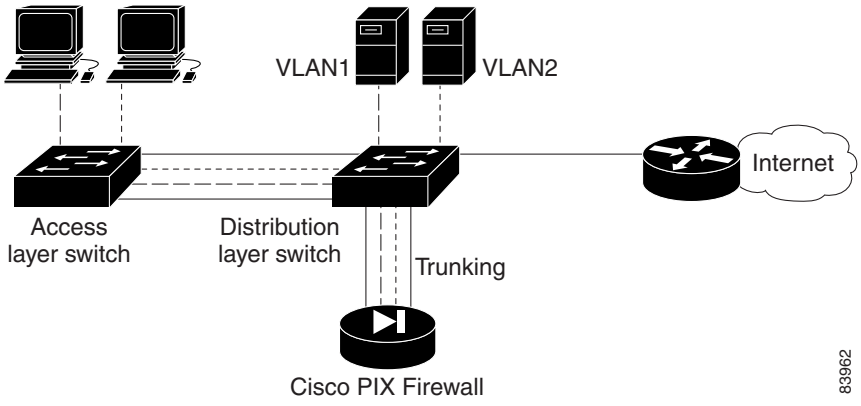
In the attack called "jumping VLANs" an attacker injects packets onto other VLANs from the native VLAN. To prevent this attack, never allow access to a native VLAN from any untrusted network. For maximum security, we recommend avoiding the use of native VLANs altogether when deploying VLANs in a secure environment. It is permitted to use native VLANs with the PIX Firewall, but you should clearly understand the security implications.

To prevent the forwarding of traffic from the PIX Firewall onto the native VLAN of a switch, use the **interface physical** command to assign a VLAN ID (other than VLAN 1) to the physical interface of the PIX Firewall. Be careful to assign a VLAN ID that is different from whatever VLAN ID is assigned to the native VLAN on the switch.

Configuring PIX Firewall with VLANs

PIX Firewall Version 6.3 introduces the capability to interconnect VLANs, as illustrated in Figure 2-9.

Figure 2-9 Using PIX Firewall (Version 6.3) to Interconnect VLANs



In Figure 2-9, two VLANs are configured on two switches. Workstations are connected to the access layer switch, while servers are connected to the distribution layer switch. Links using the 802.1q protocol interconnect the two switches and the PIX Firewall. The 802.1q protocol allows trunking VLAN traffic between devices, which means that traffic to and from multiple VLANs can be transmitted over a single physical link. Each packet contains a VLAN tag that identifies the source and destination VLAN.

The PIX Firewall supports 802.1q, allowing it to send and receive traffic for multiple VLANs on a single interface.

In Figure 2-9, the PIX Firewall is configured with one physical and one logical interface assigned to VLAN 2 and VLAN 3. The PIX Firewall interconnects the two VLANs, while providing firewall services, such as access lists, to improve network security.

To configure this example, follow these steps:

Step 1 Assign the interface speed to a physical interface by entering the following command:

```
interface ethernet0 auto
```

Step 2 Assign VLAN2 to the physical interface (ethernet0) by entering the following command:

```
interface ethernet0 vlan2 physical
```

By assigning a VLAN to the physical interface, you ensure that all frames forwarded on the interface will be tagged. VLAN 1 is not used because that is the default native VLAN for Cisco switches. Without the **physical** parameter, the default for the **interface** command is to create a logical interface.

Step 3 Create a new logical interface (VLAN3) and tie it to the physical interface (ethernet0) by entering the following command:

```
interface ethernet0 vlan3 logical
```

This will allow the PIX Firewall to send and receive VLAN-tagged packets with a VLAN identifier equal to 3 on the physical interface, ethernet0.

Step 4 Configure the logical and physical interfaces by entering the following commands:

```
nameif ethernet0 outside security0
nameif vlan3 dmz security50
ipaddress outside 192.168.101.1 255.255.255.0
ipaddress dmz 192.168.103.1 255.255.255.0
```

The first line assigns the name *outside* to ethernet0 (the physical interface) and sets the security level to zero. The second line assigns the name dmz to vlan3 (the logical interface) and sets the security level to 50. The third and fourth lines assign IP addresses to both interfaces.

After this configuration is enabled, the outside interface sends packets with a VLAN identifier of 2, and the dmz interface sends packets with a VLAN identifier of 3. Both types of packets are transmitted from the same physical interface (ethernet0).

Managing VLANs

To display information about the VLAN configuration, enter the following command:

```
show interface
```

To temporarily disable a logical interface, enter the following command:

```
interface ethernet0 vlan_id shutdown
```

Replace *vlan_id* with the VLAN ID associated with the logical interface that you want to temporarily shut down.

To change the VLAN ID of a logical interface, enter the following command:

```
interface change-vlan old_vlan_id new_vlan_id
```

Replace *old_vlan_id* with the existing VLAN ID and replace *new_vlan_id* with the new VLAN ID you want to use.

This command lets you change the VLAN ID without removing the logical interface, which is helpful if you have added a number of access-lists or firewall rules to the interface and you do not want to start over.

To disable VLAN tagging on the interface, enter the following command:

```
no interface ethernet0 vlan_id physical
```

Replace *vlan_id* with the VLAN ID for which you want to disable VLAN tagging.

To remove the logical interface and remove all configuration, enter the following command:

```
no interface ethernet0 vlan_id logical
```

Replace *vlan_id* with the VLAN ID associated with the logical interface that you want to remove.

Caution Using this command removes the interfaces and deletes all configuration rules applied to the interface.

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 5, "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

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.

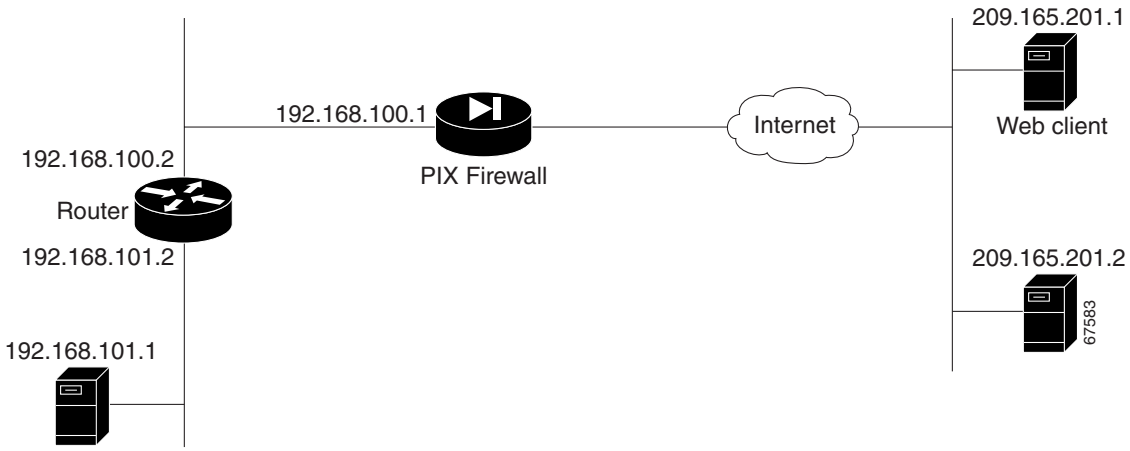
Note If outside dynamic NAT is enabled on an interface, explicit NAT policy must be configured for all hosts on the interface.

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 example, in Figure 2-10, 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 2-10 Simplifying Routing with Outside NAT



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

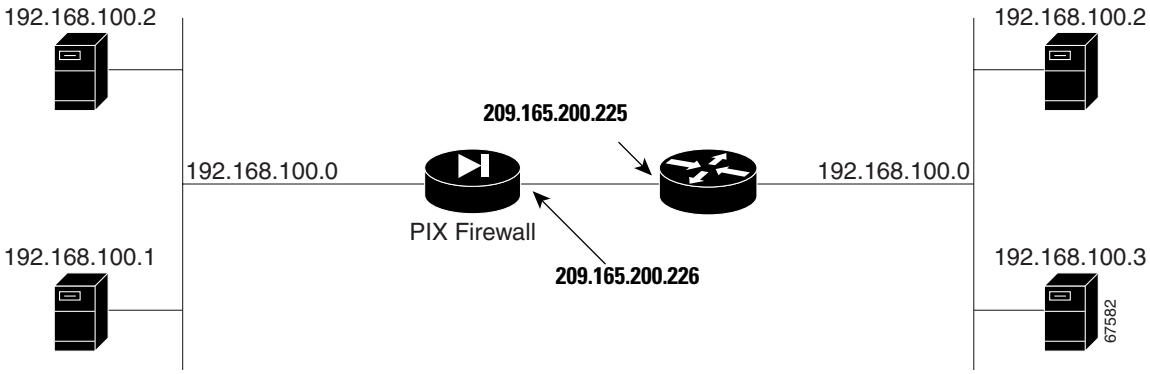
```
nat (outside) 1 209.165.201.0 255.255.255.0 outside
global (inside) 1 192.168.100.3-192.168.100.128
```

These commands translate all the source addresses on the remote network to a range of internal IP addresses (192.168.100.3-128).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 2-11, the PIX Firewall connects two private networks with overlapping address ranges.

Figure 2-11 Using Outside NAT with Overlapping Networks



In Figure 2-11, two networks use an overlapping address space and two hosts with the same IP address (192.168.100.2) must communicate. A router (209.165.200.225) connects the outside interface of the PIX Firewall (209.165.200.226) to the network on the right. The following regular NAT and outside NAT statements map each address in the private network 192.168.100.0 to the corresponding address in the public network 209.165.201.0:

```
static (inside,outside) 209.165.201.0 192.168.100.0 netmask 255.255.255.0
static (outside, inside) 209.165.201.0 192.168.100.0 netmask 255.255.255.0
```

In this example, if host 192.168.100.2 on the right network initiates a connection to host 192.168.100.2 on the left network, it uses the IP address 209.165.201.2. When the PIX Firewall receives this message, the destination address is translated from 209.165.200.2 to 192.168.100.2. Then the static that enables outside NAT is applied, and the source address is changed from 192.168.100.2 to 209.165.201.2 and is then forwarded.

The response is forwarded to the PIX Firewall with the destination address 209.165.201.2 so the outside NAT static is applied and the destination address is changed to 192.168.100.2. Then the regular NAT static is applied and the source address gets changed from 192.168.100.2 to 209.165.201.2.

Note 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 higher. 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 regular NAT, which translates the inside hosts from 192.168.100.0/24 into 209.165.201.0/24 on the outside network, is as follows:

```
static (inside,outside) 209.165.201.0 192.168.100.0 netmask 255.255.255.0
```

The NAT command for outside NAT, which translates the outside hosts from 192.168.100.0/24 into 209.165.201.0/24 on the inside network, is as follows:

```
static (outside, inside) 209.165.201.0 192.168.100.0 netmask 255.255.255.0
```

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

```
route outside 192.168.100.128 255.255.255.128 209.165.200.225 2
route outside 192.168.100.0 255.255.255.128 209.165.200.225 2
```

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

Policy NAT

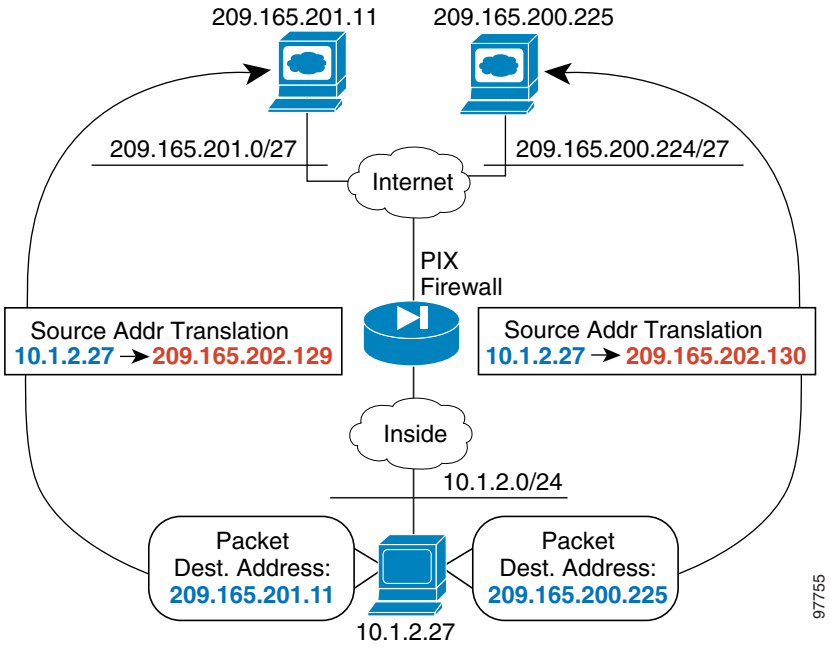
Policy NAT lets you identify local traffic for address translation by specifying the source and destination addresses (or ports) in an access list. Regular NAT uses source addresses/ports only, whereas policy NAT uses both source and destination addresses/ports.

Note All types of NAT support policy NAT, except for NAT exemption. NAT exemption uses an access list to identify the local addresses, but differs from policy NAT in that the ports are not considered.

With policy NAT, you can create multiple NAT or static statements that identify the same local address as long as the source/port and destination/port combination is unique for each statement. You can then match different global addresses to each source/port and destination/port pair.

Figure 2-12 shows a host on the 10.1.2.0/24 network accessing two different servers. When the host accesses the server at 209.165.201.11, the local address is translated to 209.165.202.129. When the host accesses the server at 209.165.200.225, the local address is translated to 209.165.202.130.

Figure 2-12 Policy NAT with Different Destination Addresses



The syntax for using global translations for the hosts shown in Figure 2-12 follows:

```
access-list NET1 permit ip 10.1.2.0 255.255.255.0 209.165.201.0 255.255.255.224
access-list NET2 permit ip 10.1.2.0 255.255.255.0 209.165.200.224 255.255.255.224
nat (inside) 1 access-list NET1
global (outside) 1 209.165.202.129 255.255.255.255
nat (inside) 2 access-list NET2
global (outside) 2 209.165.202.130 255.255.255.255
```

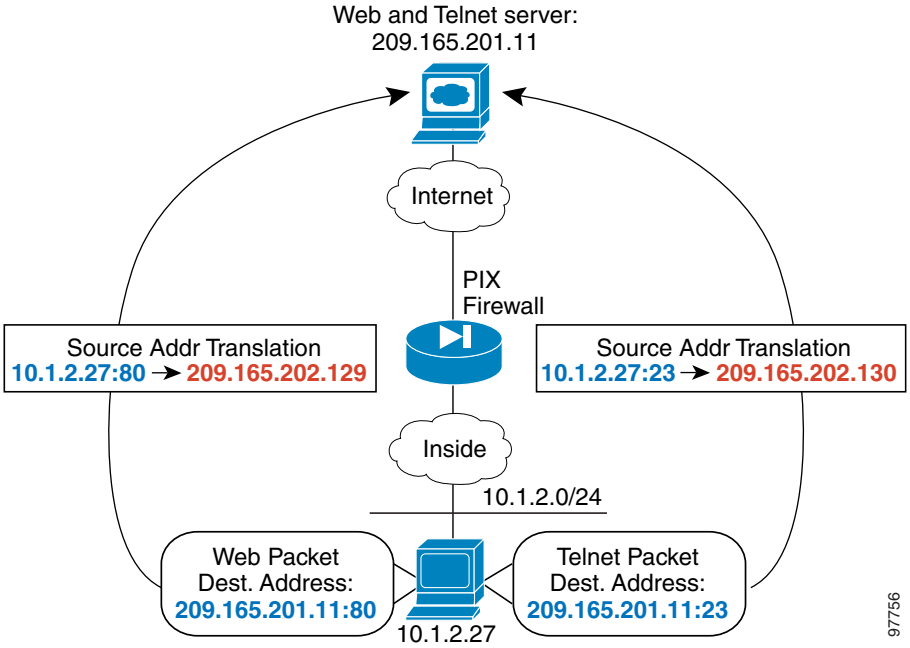
The syntax for using static translations for the two hosts shown in Figure 2-12 follows:

```
access-list NET1 permit ip host 10.1.2.27 209.165.201.0 255.255.255.224
access-list NET2 permit ip host 10.1.2.27 209.165.200.224 255.255.255.224
static (inside,outside) 209.165.202.129 access-list NET1
static (inside,outside) 209.165.202.130 access-list NET2
```

Note To prevent users from the 209.165.200.224/27 from accessing 209.165.202.129 on the PIX Firewall and to prevent users from the 209.165.201.0/27 network from accessing 209.165.202.130 on the PIX Firewall, the **ip verify reverse-path interface outside** command must be configured. This access restriction can also be enforced with ACLs applied to the outside interface without the use of the **ip verify reverse-path** command.

Figure 2-13 shows the use of source and destination ports. The host on the 10.1.2.0/24 network accesses a single host for both web services and Telnet services. When the host accesses the server for web services, the local address is translated to 209.165.202.129. When the host accesses the same server for Telnet services, the local address is translated to 209.165.202.130.

Figure 2-13 Policy NAT with Different Destination Ports



The syntax for this configuration example follows:

```
access-list WEB permit tcp 10.1.2.0 255.255.255.0 209.165.201.11 255.255.255.255 eq 80
access-list TELNET permit tcp 10.1.2.0 255.255.255.0 209.165.201.11 255.255.255.255 eq 23
nat (inside) 1 access-list WEB
global (outside) 1 209.165.202.129 255.255.255.255
nat (inside) 2 access-list TELNET
global (outside) 2 209.165.202.130 255.255.255.255
```

Limitations

The following configuration limitations apply to policy NAT:

- Access lists must contain permit statements only. Access lists for policy NAT cannot contain **deny** statements.
- An access list must be used only once with the **nat** command. For example, the following configuration would produce an error:

```
nat (inside) 1 access-list mylist-A
nat (inside) 2 access-list mylist-A
```

  Whereas, the following configuration would *not* produce an error:

```
nat (inside) 1 access-list mylist-A
nat (inside) 2 access-list mylist-B
```

- Use an access list only once between the **nat** and **static** commands.
- A global address cannot be used concurrently for NAT and PAT.
- **static** commands are matched and executed before **nat** commands.
- Policy NAT does not support SQL*Net, which is supported by regular NAT.

Configuring Policy NAT

This section describes how to configure both global translations and static translations. Refer to Figure 2-12 on page 2-42 and proceed with the configuration that fits the needs of your network.

Configuring Global Translations

Step 1 Configure IP addresses for the inside and outside interfaces.

```
ip address inside 10.1.2.1 255.255.255.0
ip address outside 209.165.202.129 255.255.255.255
```

Step 2 Configure access lists to define traffic for translation.

Note Access lists for policy NAT cannot contain **deny** statements.

```
access-list NET1 permit ip 10.1.2.0 255.255.255.0 209.165.201.0 255.255.255.224
access-list NET2 permit ip 10.1.2.0 255.255.255.0 209.165.200.224 255.255.255.224
```

Step 3 Enter **nat** commands that use the same identifier as those defined with the **access-list** statements in Step 2.

```
nat (inside) 1 access-list NET1
nat (inside) 2 access-list NET2
```

Step 4 Enter **global** commands to associate the outside addresses for translation to the outside destination networks.

```
global (outside) 1 209.165.202.129 255.255.255.255
global (outside) 2 209.165.202.130 255.255.255.255
```

Configuring Static Translations

Step 1 Configure IP addresses for the inside and outside interfaces.

```
ip address inside 10.1.2.1 255.255.255.0
ip address outside 209.165.202.129 255.255.255.255
```

Step 2 Configure access lists to define traffic for translation.

Note Access lists for policy NAT cannot contain **deny** statements.

```
access-list NET1 permit ip host 10.1.2.27 209.165.201.0 255.255.255.224
access-list NET2 permit ip host 10.1.2.27 209.165.200.224 255.255.255.224
```

Step 3 Configure static translations to individual hosts.

```
static (inside,outside) 209.165.202.129 access-list NET1
static (inside,outside) 209.165.202.130 access-list NET2
```

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

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.

Enter the following subcommand for each multicast interface, which is typically an inside or more secure interface.

```
igmp forward interface mc-source-if-name
```

Replace *mc-source-if-name* with the name of the PIX Firewall interface that is connected to the MC router. This is typically the outside interface. For example, the following command enables the forwarding of IGMP reports on the currently selected PIX Firewall interface, when the MC router is connected to the interface named "outside."

```
igmp forward interface outside
```

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

```
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 toward the receiving hosts.

Step 5 Create an access list entry to permit inbound traffic to the multicast address:

```
access-list acl_ID permit udp host ip-address host group-address
```

Step 6 Apply the access list to the Outside interface for inbound multicast transmissions:

```
access-group acl_ID in interface outside
```

Note It is suggested that you narrow down the host that is sourcing the multicast stream.

Step 7 (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 8 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 multicast interface that you are currently configuring.

Example 2-6 Inside Receiving Hosts

In the following example, inside clients must register with the multicast group with the Class D address 225.2.1.14:

```
multicast interface inside
  igmp join-group 225.2.1.14
```

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 2-7 Inside Receiving Hosts with Access Control

The following example configures the inside and DMZ receivers:

```
multicast interface outside
    igmp access-group 1
multicast interface inside
    igmp forward interface outside
    igmp access-group 1
multicast interface dmz
    igmp forward interface outside
    igmp access-group 1
! The following permits igmp messages to 225.2.1.0/25 network
access-list 1 permit igmp any 225.2.1.0 255.255.255.128
access-list 1 deny ip any any

! The following permits multicast packets in the network 225.2.1.0/25 in the
! outside interface of the PIX
access-list 100 permit udp any 225.2.1.0 255.255.255.128
access-list 100 in interface outside
```

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 2-8 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, page 2-49
- Setting Query Response Time, page 2-49

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. The following is sample output for a working configuration:

```
pix-2(config)# show igmp
IGMP is enabled on interface outside
  IGMP querying router is 192.168.9.1
IGMP Connected Group Membership
   Group Address    Interface            Uptime    Expires   Last Reporter
IGMP is enabled on interface inside
  Current IGMP version is 2
  IGMP query interval is 60 seconds
  IGMP querier timeout is 125 seconds
  IGMP max query response time is 10 seconds
  Last member query response interval is 1 seconds
  Inbound IGMP access group is 1
  IGMP max groups is 500
  IGMP activity: 1 joins, 0 leaves
  IGMP forwarding on interface outside
  IGMP querying router is 10.10.10.161 (this system)
IGMP Connected Group Membership
   Group Address    Interface            Uptime    Expires   Last Reporter
   225.2.1.14       inside               19:10:41  never     10.10.10.161
```

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. The following is sample output for a working configuration:

```
pix-2(config)# show mroute
IP Multicast Forwarding Information Base
Entry flags: C - Directly-Connected Check, S - Signal, D - Drop
Interface flags: F - Forward, A - Accept, IC - Internal Copy,
NS - Negate Signal, DP - Don't Preserve, SP - Signal Present,
```

```
EG - Egress
Forwarding Counts: Packets in/Packets out/Bytes out
Failure Counts: RPF / TTL / Empty Olist / Other
(*,225.2.1.14),  Flags: S
  Last Used: 0:00:16
  Forwarding Counts: 3/1/188
  Failure Counts: 0/0/2/0
  inside Flags: F
(192.168.1.113,225.2.1.14),  Flags:
  Last Used: 0:00:00
  Forwarding Counts: 1128/1128/212064
  Failure Counts: 0/0/0/0
  outside Flags: A SP
  inside Flags: F
```

The following is sample output from the **show mroute** command for a non-working configuration:

```
pix-2(config)# show mroute
IP Multicast Forwarding Information Base
Entry flags: C - Directly-Connected Check, S - Signal, D - Drop
Interface flags: F - Forward, A - Accept, IC - Internal Copy,
NS - Negate Signal, DP - Don't Preserve, SP - Signal Present,
EG - Egress
Forwarding Counts: Packets in/Packets out/Bytes out
Failure Counts: RPF / TTL / Empty Olist / Other
(*,225.2.1.14),  Flags: S
  Last Used: 0:02:18
  Forwarding Counts: 4/1/188
  Failure Counts: 0/0/3/0
  inside Flags: F
(192.168.1.113,225.2.1.14),  Flags:
  Last Used: 17:57:09
  Forwarding Counts: 502/0/0
  Failure Counts: 0/0/502/0
  outside Flags: A SP
  inside Flags: F
```

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 mfwd
```

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/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