



## Command Reference

---

This chapter provides detailed descriptions on most of the PIX Firewall commands.

**Note**

---

The IPsec-related commands are described in the “Command Reference” chapter of the *IPsec User Guide for the Cisco Secure PIX Firewall Version 5.2*.

---

Before reading the PIX Firewall “Command Reference” chapter, read the following:

- Chapter 1, “Introduction,” for command line guidelines and ports and protocols information.
- Chapter 2, “Configuring the PIX Firewall,” to configure PIX Firewall and test connectivity.
- *IPsec User Guide for the Cisco Secure PIX Firewall Version 5.2* for background information about IPsec and its components, and how to implement these IPsec features in the PIX Firewall to create a Virtual Private Network (VPN).

The following notes can help you as you configure the PIX Firewall:

- View your configuration at any time with the **write terminal** command.
- Save your configuration frequently with the **write memory** command.
- Always check the syntax before entering a command. Enter a command and press the Enter key to view a quick summary, or precede a command with **help**, as in, **help aaa**.
- View syslog messages as you work on the PIX Firewall. Start accumulating messages with the **logging buffered debugging** command, view messages with the **show logging** command, and clear the message buffer with the **clear logging** command. Syslog messages are described in *System Log Messages for the Cisco Secure PIX Firewall Version 5.2*.
- PIX Firewall documentation is available online at the following site:  
<http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix>
- Abbreviate commands, such as, using the **con te** command statement to start configuration mode, the **wr t** command statement to list the configuration, and **wr m** to write to Flash memory. Start logging with the **lo b 7** command statement and show logging messages with the **sh lo** command statement.
- After changing or removing the **alias**, **access-list**, **conduit**, **global**, **nat**, **outbound**, and **static** commands, use the **clear xlate** command to make the IP addresses available for access. If traffic is not moving correctly, reboot the PIX Firewall.
- You can view possible port and protocol numbers at the following IANA web sites:  
<http://www.isi.edu/in-notes/iana/assignments/port-numbers>  
<http://www.isi.edu/in-notes/iana/assignments/protocol-numbers>
- Create your configuration on a text editor and cut and paste it into the configuration. PIX Firewall lets you paste in a line at a time or the whole configuration. Always check your configuration after pasting large blocks of text to be sure everything copied.

# aaa

Enable, disable, or view TACACS+ or RADIUS user authentication, authorization, and accounting for the server previously designated with the **aaa-server** command. (Configuration mode.)

**aaa accounting include | exclude** *acctg\_service* **inbound | outbound** | *if\_name local\_ip local\_mask foreign\_ip foreign\_mask group\_tag*

**no aaa accounting include | exclude** *authn\_service* **inbound | outbound** | *if\_name group\_tag*

**aaa accounting match** *acl\_name* **inbound | outbound** | *if\_name group\_tag*

**no aaa accounting match** *acl\_name* **inbound | outbound** | *if\_name group\_tag*

**aaa authentication include | exclude** *authn\_service* **inbound | outbound** | *if\_name local\_ip local\_mask foreign\_ip foreign\_mask group\_tag*

**no aaa authentication [include | exclude** *authn\_service* **inbound | outbound** | *if\_name local\_ip local\_mask foreign\_ip foreign\_mask group\_tag*]

**aaa authentication match** *acl\_name* **inbound | outbound** | *if\_name group\_tag*

**no aaa authentication match** *acl\_name* **inbound | outbound** | *if\_name group\_tag*

**aaa authentication [serial | enable | telnet | ssh] console** *group\_tag*

**no aaa authentication [serial | enable | telnet | ssh] console** *group\_tag*

**aaa authorization include | exclude** *author\_service* **inbound | outbound** | *if\_name local\_ip local\_mask foreign\_ip foreign\_mask*

**no aaa authorization [include | exclude** *author\_service* **inbound | outbound** | *if\_name local\_ip local\_mask foreign\_ip foreign\_mask*]

**aaa authorization match** *acl\_name* **inbound | outbound** | *if\_name group\_tag*

**no aaa authorization match** *acl\_name* **inbound | outbound** | *if\_name group\_tag*

**clear aaa [accounting include | exclude** *authn\_service* **inbound | outbound** | *if\_name group\_tag*]

**clear aaa [authentication include | exclude** *authn\_service* **inbound | outbound** | *if\_name local\_ip local\_mask foreign\_ip foreign\_mask group\_tag*]

**clear aaa [authorization [include | exclude** *author\_service* **inbound | outbound** | *if\_name local\_ip local\_mask foreign\_ip foreign\_mask*]]

**show aaa**

## Syntax Description

<b>accounting</b>	Enable or disable accounting services with authentication server. Use of this command requires that you previously used the <b>aaa-server</b> command to designate an authentication server.
<b>include</b>	Create a new rule with the specified service to include.
<b>exclude</b>	Create an exception to a previously stated rule by excluding the specified service from authentication, authorization, or accounting to the specified host. The <b>exclude</b> parameter improves the former <b>except</b> option by allowing the user to specify a port to exclude to a specific host or hosts.
<i>acctg_service</i>	<p>The accounting service. Accounting is provided for all services or you can limit it to one or more services. Possible values are <b>any</b>, <b>ftp</b>, <b>http</b>, <b>telnet</b>, or <i>protocol/port</i>. Use <b>any</b> to provide accounting for all TCP services. To provide accounting for UDP services, use the <i>protocol/port</i> form.</p> <p>For <i>protocol/port</i>, the TCP <i>protocol</i> appears as 6, the UDP protocol appears as 17, and so on, and port is the TCP or UDP destination port. A port value of 0 (zero) means all ports. For protocols other than TCP and UDP, the <i>port</i> is not applicable and should not be used.</p>
<b>match</b> <i>acl_name</i>	Specify an <b>access-list</b> command statement name.
<b>authentication</b>	<p>Enable or disable user authentication, prompt user for username and password, and verify information with authentication server.</p> <p>When used with the <b>console</b> option, enables or disables authentication service for access to the PIX Firewall console over Telnet or from the Console connector on the PIX Firewall unit.</p> <p>Use of the <b>aaa authentication</b> command requires that you previously used the <b>aaa-server</b> command to designate an authentication server.</p>
<i>authen_service</i>	<p>The application with which a user is accessing a network. Use <b>any</b>, <b>ftp</b>, <b>http</b>, or <b>telnet</b>. The <b>any</b> value enables accounting or authentication for all TCP services. To have users prompted for authentication credentials, they must use FTP, HTTP, or Telnet. (HTTP is the Web and only applies to web browsers that can prompt for a username and password.)</p> <p>If the authentication or authorization server is authenticating services other than FTP, HTTP, or Telnet, using <b>any</b> will not permit those services to authenticate in the firewall. The firewall only knows how to communicate with FTP, HTTP, and Telnet for authentication and authorization.</p> <p>Only set this parameter to a service other than <b>any</b> if the authentication or authorization server is set the same way. Unless you want to temporarily restrict access to a specific service, setting a service in this command can increase system administration work and may cause all connections to fail if the authentication or authorization server is authenticating one service and you set this command to another.</p>

**authorization** Enable or disable TACACS+ user authorization for services (PIX Firewall does not support RADIUS authorization). The authentication server determines what services the user is authorized to access.

*author\_service* The services which require authorization. Use **any**, **ftp**, **http**, **telnet**, or *protocol/port*. Services not specified are authorized implicitly. Services specified in the **aaa authentication** command do not affect the services which require authorization.

For *protocol/port*:

- *protocol*—the protocol (6 for TCP, 17 for UDP, 1 for ICMP, and so on).
- *port*—the TCP or UDP destination port, or port range. The *port* can also be the ICMP type; that is, 8 for ICMP echo or ping. A port value of 0 (zero) means all ports. Port ranges only applies to the TCP and UDP protocols, not to ICMP. For protocols other than TCP, UDP, and ICMP the *port* is not applicable and should not be used. An example port specification follows:

```
aaa authorization include udp/53-1024 inside 0 0 0 0
```

This example enables authorization for DNS lookups to the inside interface for all clients, and authorizes access to any other services that have ports in the range of 53 to 1024.



**Note** Specifying a port range may produce unexpected results at the authorization server. PIX Firewall sends the port range to the server as a string with the expectation that the server will parse it out into specific ports. Not all servers do this. In addition, you may want users to be authorized on specific services, which will not occur if a range is accepted.

**inbound** Authenticate or authorize inbound connections. Inbound means the connection originates on the outside interface and is being directed to the inside interface.

**outbound** Authenticate or authorize outbound connections. Outbound means the connection originates on the inside and is being directed to the outside interface.

*if\_name* Interface name from which users require authentication. Use *if\_name* in combination with the *local\_ip* address and the *foreign\_ip* address to determine where access is sought and from whom. The *local\_ip* address is always on the highest security level interface and *foreign\_ip* is always on the lowest. See the Examples section for how the *if\_name* affects the use of this command.

*local\_ip* The IP address of the host or network of hosts that you want to be authenticated or authorized. You can set this address to **0** to mean all hosts and to let the authentication server decide which hosts are authenticated.

*local\_mask* Network mask of *local\_ip*. Always specify a specific mask value. Use 0 if the IP address is 0. Use 255.255.255.255 for a host.

*foreign\_ip* The IP address of the hosts you want to access the *local\_ip* address. Use 0 to mean all hosts.

*foreign\_mask* Network mask of *foreign\_ip*. Always specify a specific mask value. Use 0 if the IP address is 0. Use 255.255.255.255 for a host.

### **console**

Specify that access to the PIX Firewall console require authentication and optionally, log configuration changes to a syslog server.

The **aaa authentication serial console** command lets you require authentication verification to access the PIX Firewall unit's serial console. The **serial console** options also logs to a syslog server changes made to the configuration from the serial console.

Authenticated access to the PIX Firewall console has different types of prompts depending on the option you choose with the **aaa authentication [serial | enable | telnet | ssh] console** command. While the **enable** and **ssh** options allow three tries before stopping with an access denied message, both the **serial** and **telnet** options cause the user to be prompted continually until successfully logging in. The **serial** option requests a username and password before the first command line prompt on the serial console connection. The **telnet** option forces you to specify a username and password before the first command line prompt of a Telnet console connection. The **enable** option requests a username and password before accessing privileged mode for serial, Telnet, or SSH connections. The **ssh** option requests a username and password before the first command line prompt on the SSH console connection. The **ssh** option allows a maximum of three authentication attempts.

Telnet access to the PIX Firewall console is available from any internal interface, and from the outside interface with IPSec configured, and requires previous use of the **telnet** command. SSH access to the PIX Firewall console is also available from any interface without IPSec configured, and requires previous use of the **ssh** command.

The new **ssh** option specifies the group of AAA servers to be used for SSH user authentication. The authentication protocol and AAA server IP addresses are defined with the **aaa-server** command statement.

Similar to the Telnet model, if an **aaa authentication ssh console group\_tag** command statement is not defined, you can gain access to the PIX Firewall console with the username **pix** and with the PIX Firewall Telnet password (set with the **passwd** command). If the **aaa** command is defined but the SSH authentication requests a timeout, which implies the AAA servers may be down or not available, you can gain access to the PIX Firewall using username **pix** and the enable password (set with the **enable password** command). By default, the Telnet password is **cisco** and the enable password is not set.

If the console login request times out, you can gain access to the PIX Firewall from the serial console by entering the **pix** username and the enable password.

The maximum password length for accessing the console is 16 characters.

### *group\_tag*

The group tag set with the **aaa-server** command.

### Usage Guidelines

The **aaa** command enables or disables the following AAA (Authentication, Authorization, and Accounting) features:

- User authentication services. A user starting a connection via FTP, Telnet, or over the World Wide Web is prompted for their username and password. An authentication server, designated previously with the **aaa-server** command, verifies whether the username and password are correct. If the username and password are correct, PIX Firewall lets further traffic between the authentication server and the connection interact independently through the PIX Firewall unit's "Cut-Through Proxy" feature.
- Authentication access to the PIX Firewall unit's console via Telnet, SSH, or the serial console. (Telnet access requires previous use of the **telnet** command. SSH access requires previous use of the **ssh** command.)
- User authorization services for TACACS+ connections that let the authentication server determine which services the user can access.
- Accounting services so that administrators can track which hosts accessed the PIX Firewall. AAA accounting can also track FTP/Telnet/HTTP connections initiated with IPSec.



**Note** RADIUS authorization is supported with the use of **access-list** command statement and configuring a RADIUS server to send an **acl=acl\_name** vendor-specific identifier. Refer to the **access-list** command page for more information.



**Note** PIX Firewall listens for RADIUS on ports 1645 and 1646. If your RADIUS server uses ports 1812 and 1813, you will need to reconfigure it to use ports 1645 and 1646.



**Note** If the AAA console login request times out, you can gain access to the PIX Firewall from the serial console by entering the **pix** username and the enable password.

### match *acl\_name* Option Usage

The syntax for this command is as follows:

```
aaa authentication | authorization | accounting match acl_name inbound | outbound |
interface_name group_tag
```

An example is as follows:

```
show access-list
access-list mylist permit tcp 10.0.0.0 255.255.255.0 172.23.2.0 255.255.255.0 (hitcnt=0)
access-list yourlist permit tcp any any (hitcnt=0)
show aaa
aaa authentication match mylist outbound TACACS+
```

Similar to IPSec, the keyword **permit** means "yes" and **deny** means "no." Therefore, the following command:

```
aaa authentication match yourlist outbound tacacs
```

is equal to this command:

```
aaa authentication include any outbound 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 tacacs
```

The **aaa** command statement list is order dependent between **access\_list** command statements. If the following command is entered:

```
aaa authentication match yourlist outbound tacacs
```

after this command:

```
aaa authentication match mylist outbound TACACS+
```

PIX Firewall tries to find a match in the mylist **access-list** command statement group before it tries to find a match in the yourlist **access-list** command statement group.

Old **aaa** command configuration and functionality stays the same and is not converted to the **access\_list** format. Hybrid configurations; that is, old configurations combined with the new **access\_list** configuration are not recommended.

### Usage Notes

1. The maximum username prompt for HTTP authentication is 30 characters. The maximum password length is 15 characters.
2. The **aaa** command is not intended to mandate your security policy. The authentication and authorization servers determine whether a user can or cannot access the system, what services can be accessed, and what IP addresses the user can access. The PIX Firewall interacts with FTP, HTTP (Web access), and Telnet to display the credentials prompts for logging in to the network or logging in to exit the network. You can specify that only a single service be authenticated, but this must agree with the authentication server to ensure that both the firewall and server agree.
3. Accounting information is only sent to the active server in a server group.
4. The new **include** and **exclude** options are not backward compatible with previous PIX Firewall versions. If you downgrade to an earlier version, the **aaa** command statements will be removed from your configuration.
5. The prompts users see requesting AAA credentials differ between the three services that can access the PIX Firewall for authentication: Telnet, FTP, and HTTP (Web):
  - a. Telnet users see a prompt generated by the PIX Firewall that you can change with the **auth-prompt** command. The PIX Firewall permits a user up to four chances to log in and then if the username or password still fails, the PIX Firewall drops the connection.
  - b. FTP users receive a prompt from the FTP program. If a user enters an incorrect password, the connection is dropped immediately. If the username or password on the authentication database differs from the username or password on the remote host to which you are using FTP to access, enter the username and password in these formats:

```
authentication_user_name@remote_system_user_name  
authentication_password@remote_system_password
```

If you daisy-chain PIX Firewall units, Telnet authentication works in the same way as a single unit, but FTP and HTTP authentication have additional complexity for users because they have to enter each password and username with an additional at (@) character and password or username for each daisy-chained system. Users can exceed the 63-character password limit depending on how many units are daisy-chained and password length.

Some FTP graphical user interfaces (GUIs) do not display challenge values.

- c. HTTP users see a pop-up window generated by the browser itself. If a user enters an incorrect password, the user is reprompted. When the web server and the authentication server are on different hosts, use the **virtual** command to get the correct authentication behavior.

6. Use of the **aaa authorization** command requires previous use of the **aaa authentication** command; however, use of the **aaa authentication** command does not require use of an **aaa authorization** command.
7. If you want to allow connections to come from any host, code the local IP address and netmask as **0.0.0.0 0.0.0.0**, or **0 0**. The same convention applies to the foreign host IP address and netmask; **0.0.0.0 0.0.0.0** means any foreign host.
8. Authenticated access to the PIX Firewall console has different types of prompts depending on the option you choose with the **aaa authentication ... console** command:
  - a. **enable** option—Allows three tries before stopping with “Access denied.” The **enable** option requests a username and password before accessing privileged mode for serial or Telnet connections.
  - b. **serial** option—Causes the user to be prompted continually until successfully logging in. The **serial** option requests a username and password before the first command line prompt on the serial console connection.
  - c. **telnet** option—Causes the user to be prompted continually until successfully logging in. The **telnet** option forces you to specify a username and password before the first command line prompt of a Telnet console connection.
9. You can specify an interface name with **aaa authentication**. In previous versions, if you specified **aaa authentication include any outbound 0 0 server**, PIX Firewall only authenticated outbound connections and not those to the perimeter interface. PIX Firewall now authenticates any outbound connection to the outside as well as to hosts on the perimeter interface. To preserve the behavior of previous versions, use these commands to enable authentication and to disable authentication from the inside to the perimeter interface:

```
aaa authentication include any outbound 0 0 server
aaa authentication exclude outbound perim_net perim_mask server
```

10. When using HTTP authentication to a site running Microsoft IIS that has “Basic text authentication” or “NT Challenge” enabled, users may be denied access from the Microsoft IIS server. This occurs because the browser appends the string: “Authorization: Basic=Uuhjksdkfhk==” to the HTTP GET commands. This string contains the PIX Firewall authentication credentials.

Windows NT Microsoft IIS servers respond to the credentials and assume that a Windows NT user is trying to access privileged pages on the server. Unless the PIX Firewall username password combination is exactly the same as a valid Windows NT username and password combination on the Microsoft IIS server, the HTTP GET command is denied.

To solve this problem, PIX Firewall provides the **virtual http** command which redirects the browser's initial connection to another IP address, authenticates the user, then redirects the browser back to the URL which the user originally requested.

Once authenticated, a user never has to reauthenticate no matter how low the PIX Firewall uauth timeout is set. This is because the browser caches the “Authorization: Basic=Uuhjksdkfhk==” string in every subsequent connection to that particular site. This can *only* be cleared when the user exits *all* instances of Netscape Navigator or Internet Explorer and restarts. Flushing the cache is of no use.

As long as the user repeatedly browses the Internet, the browser resends the “Authorization: Basic=Uuhjksdkfhk==” string to transparently reauthenticate the user.

11. Multimedia applications such as CU-SeeMe, InternetPhone, MeetingPoint, and MS Netmeeting silently start the HTTP service before an H.323 session is established from the inside to the outside.

To avoid interfering with these applications, do not enter blanket outgoing AAA command statements for all challenged ports such as using the **any** option. Be selective with which ports and addresses you use to challenge HTTP, and when to set user authentication timeouts to a higher timeout value. If interfered with, the multimedia programs may fail on the PC and may even crash the PC after establishing outgoing sessions from the inside.

12. For outbound connections, first use the **nat** command to determine which IP addresses can access the firewall. For inbound connections, first use the **static** and **access-list** command statements to determine which inside IP addresses can be accessed through the firewall from the outside network.
13. When a host is configured for authentication, all users on the host have to use a web browser or Telnet first before performing any other networking activity, such as accessing mail or a news reader. The reason for this is that users must first establish their authentication credentials and programs such as mail agents and newsreaders do not have authentication challenge prompts.
14. The PIX Firewall only accepts 7-bit characters during authentication. After authentication, the client and server can negotiate for 8-bits if required. During authentication, the PIX Firewall only negotiates Go-Ahead, Echo, and NVT (network virtual terminal).
15. Up to 256 TACACS+ or RADIUS servers are permitted (up to 16 servers in each of the up to 16 server groups—set with the **aaa-server** command). When a user logs in, the servers are accessed one at a time starting with the first server you specify in the configuration, until a server responds.
16. For each IP address, one **aaa authentication** command is permitted for inbound connections and one for outbound connections. Also, for an IP address, one **aaa authorization** command is permitted. If you want to authorize more than one service with **aaa authorization**, use the **any** parameter for the service type.
17. The PIX Firewall permits only one authentication type per network. For example, if one network connects through the PIX Firewall using TACACS+ for authentication, another network connecting through the PIX Firewall can authenticate with RADIUS, but one network cannot authenticate with both TACACS+ and RADIUS.
18. For the TACACS+ server, if you do not specify a key to the **aaa-server** command, no encryption occurs.
19. Network browsers such as Netscape Navigator do not present a challenge value during authentication; therefore, only password authentication can be used from a network browser.
20. PIX Firewall supports authentication usernames up to 127 characters and passwords of up to 63 characters. A password or username may not contain an at (@) character as part of the password or username string, except as shown in Note 5.
21. The PIX Firewall displays the same timeout message for both RADIUS and TACACS+. The message “aaa server host machine not responding” displays when either of the following occurs:
  - a. The AAA server system is down.
  - b. The AAA server system is up, but the service is not running.Previously, TACACS+ differentiated the two states above and provided two different timeout messages, while RADIUS did not differentiate the two states and provided one timeout message.
22. If the first attempt at authorization fails and a second attempt causes a timeout, use the **service resetinbound** command to reset the client that failed the authorization so that it will not retransmit any connections. An example authorization timeout message in Telnet follows:

```
Unable to connect to remote host: Connection timed out
```

See also: **aaa-server**, **auth-prompt**, **service**, **ssh**, **telnet**, **virtual**.

## Examples

1. The following example lists the new include and exclude options:

```
aaa authentication include any outbound 172.31.0.0 255.255.0.0 0.0.0.0 0.0.0.0
tacacs+
aaa authentication exclude telnet outbound 172.31.38.0 255.255.255.0 0.0.0.0 0.0.0.0
tacacs+
```

2. The following examples demonstrate ways to use the *if\_name* parameter. The PIX Firewall has an inside network of 192.168.1.0, an outside network of 209.165.201.0 (subnet mask 255.255.255.224), and a perimeter network of 209.165.202.128 (subnet mask 255.255.255.224).

This example enables authentication for connections originated from the inside network to the outside network:

```
aaa authentication include any outbound 192.168.1.0 255.255.255.0 209.165.201.0
255.255.255.224 tacacs+
```

This example enables authentication for connections originated from the inside network to the perimeter network:

```
aaa authentication include any outbound 192.168.1.0 255.255.255.0 209.165.202.128
255.255.255.224 tacacs+
```

This example enables authentication for connections originated from the outside network to the inside network:

```
aaa authentication include any inbound 192.168.1.0 255.255.255.0 209.165.201.0
255.255.255.224 tacacs+
```

This example enables authentication for connections originated from the outside network to the perimeter network:

```
aaa authentication include any inbound 209.165.201.0 255.255.255.224 209.165.202.128
255.255.255.224 tacacs+
```

This example enables authentication for connections originated from the perimeter network to the outside network:

```
aaa authentication include any perimeter 209.165.202.128 255.255.255.224
209.165.201.0 255.255.255.224 tacacs+
```

3. This example specifies that IP addresses 10.0.0.1 through 10.0.0.254 can originate outbound connections and then enables user authentication so that those addresses must enter user credentials to exit the firewall. In this example, the first **aaa authentication** command permits authentication on FTP, HTTP, or Telnet depending on what the authentication server handles. The second **aaa authentication** command lets host 10.0.0.42 start outbound connections without being authenticated. This example uses the default authentication group **tacacs+**:

```
nat (inside) 1 10.0.0.0 255.255.255.0
aaa authentication include any outbound 0 0 tacacs+
aaa authentication exclude outbound 10.0.0.42 255.255.255.255 tacacs+ any
```

- This example permits inbound access to any IP address in the range of 209.165.201.1 through 209.165.201.30 indicated by the 209.165.201.0 network address (subnet mask 255.255.255.224). All services are permitted by the **access-list** command, and the **aaa authentication** command permits authentication on FTP, HTTP, or Telnet depending on what the authentication server handles. The authentication server is at IP address 10.16.1.20 on the inside interface:

```
aaa-server AuthIn protocol tacacs+
aaa-server AuthIn (inside) host 10.16.1.20 thisisakey timeout 20
static (inside,outside) 209.165.201.0 10.16.1.0 netmask 255.255.255.224
access-list acl_out permit tcp 10.16.1.0 255.255.255.0 209.165.201.0 255.255.255.224
access-group acl_out in interface outside
aaa authentication include any inbound 0 0 AuthIn
```

- This example enables authorization for DNS lookups from the outside interface:

```
aaa authorization include udp/53 inbound 0.0.0.0 0.0.0.0
```

- This example enables authorization of ICMP echo-reply packets arriving at the inside interface from inside hosts:

```
aaa authorization include 1/0 outbound 0.0.0.0 0.0.0.0
```

This means that users will not be able to ping external hosts if they have not been authenticated using Telnet, HTTP, or FTP.

- This example enables authorization for ICMP echoes (pings) only that arrive at the inside interface from an inside host:

```
aaa authorization include 1/8 outbound 0.0.0.0 0.0.0.0
```

# aaa-server

Specify an AAA server. (Configuration mode.)

**aaa-server** *group\_tag* (*if\_name*) **host** *server\_ip* **key** **timeout** *seconds*

**no aaa-server** *group\_tag* (*if\_name*) **host** *server\_ip* **key** **timeout** *seconds*

**aaa-server** *group\_tag* **protocol** *auth\_protocol*

**clear aaa-server** [*group\_tag*]

**show aaa-server**

## Syntax Description

<i>group_tag</i>	An alphanumeric string which is the name of the server group. Use the <i>group_tag</i> in the <b>aaa</b> command to associate <b>aaa authentication</b> and <b>aaa accounting</b> command statements to an AAA server.
<i>if_name</i>	The interface name on which the server resides.
<b>host</b> <i>server_ip</i>	The IP address of the TACACS+ or RADIUS server.
<i>key</i>	A case-sensitive, alphanumeric keyword of up to 127 characters that is the same value as the key on the TACACS+ server. Any characters entered past 127 are ignored. The key is used between the client and server for encrypting data between them. The <i>key</i> must be the same on both the client and server systems. Spaces are not permitted in the key, but other special characters are.
<b>timeout</b> <i>seconds</i>	A retransmit timer that specifies the duration that the PIX Firewall retries access four times to the AAA server before choosing the next AAA server. The default is 5 seconds. The maximum time is 30 seconds.  For example, if the timeout value is 10 seconds, PIX Firewall retransmits for 10 seconds and if no acknowledgment is received, tries three times more for a total of 40 seconds to retransmit data before the next AAA server is selected.
<b>protocol</b> <i>auth_protocol</i>	The type of AAA server, either <b>tacacs+</b> or <b>radius</b> .

## Usage Guidelines

The **aaa-server** command lets you specify an AAA server group. PIX Firewall lets you define separate groups of TACACS+ or RADIUS servers for specifying different types of traffic; such as, a TACACS+ server for inbound traffic and another for outbound traffic. Another use is where all outbound HTTP traffic will be authenticated by a TACACS+ server, and all inbound traffic will use RADIUS.

AAA server group are defined by a tag name that directs different types of traffic to each authentication server. If the first authentication server in the list fails, the AAA subsystem fails over to the next server in the tag group. You can have up to 14 tag groups and each group can have up to 14 AAA servers for a total of up to 196 AAA servers.

The **aaa** command references the tag group.

**Note**

The previous server type option at the end of the **aaa authentication** and **aaa accounting** commands has been replaced with the **aaa-server** group tag. Backward compatibility with previous versions is maintained by the inclusion of two default protocols for TACACS+ and RADIUS.

If accounting is in effect, the accounting information goes only to the active server.

The default configuration provides these two **aaa-server** protocols:

```
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
```

**Note**

If you are upgrading from a previous version of PIX Firewall and have **aaa** command statements in your configuration, using the default server groups lets you maintain backward compatibility with the **aaa** command statements in your configuration.

**Examples**

1. This example uses the default protocol **tacacs+** with the **aaa** commands:

```
aaa-server TACACS+ (inside) host 10.1.1.10 thekey timeout 20
aaa authentication include any outbound 0 0 0 0 TACACS+
aaa authorization include any outbound 0 0 0 0
aaa accounting include any outbound 0 0 0 0 TACACS+
aaa authentication serial console TACACS+
```

This example specifies that the authentication server with the IP address 10.1.1.10 resides on the inside interface and is in the default TACACS+ server group. The next three command statements specify that any users starting outbound connections to any foreign host will be authenticated using TACACS+, that the users who are successfully authenticated are authorized to use any service, and that all outbound connection information will be logged in the accounting database. The last command statement specifies that access to the PIX Firewall unit's serial console requires authentication from the TACACS+ server.

2. This example creates the AuthOut and AuthIn server groups for RADIUS authentication and specifies that servers 10.0.1.40, 10.0.1.41, and 10.1.1.2 on the inside interface provide authentication. The servers in the AuthIn group authenticate inbound connections, the AuthOut group authenticates outbound connections:

```
aaa-server AuthIn protocol radius
aaa-server AuthIn (inside) host 10.0.1.40 ab timeout 20
aaa-server AuthIn (inside) host 10.0.1.41 abc timeout 4
aaa-server AuthOut protocol radius
aaa-server AuthOut (inside) host 10.1.1.2 abc123 timeout 15
aaa authentication include any inbound 0 0 0 0 AuthIn
aaa authentication include any outbound 0 0 0 0 AuthOut
```

- This example lists the commands that can be used to establish an Xauth crypto map:

```
ip address inside 10.0.0.1 255.255.255.0
ip address outside 168.20.1.5 255.255.255.0
ip local pool dealer 10.1.2.1-10.1.2.254
nat (inside) 0 access-list 80
aaa-server TACACS+ host 10.0.0.2 secret123
crypto ipsec transform-set pc esp-des esp-md5-hmac
crypto dynamic-map cisco 4 set transform-set pc
crypto map partner-map 20 ipsec-isakmp dynamic cisco
crypto map partner-map client configuration address initiate
crypto map partner-map client authentication TACACS+
crypto map partner-map interface outside
isakmp key cisco1234 address 0.0.0.0 netmask 0.0.0.0
isakmp client configuration address-pool local dealer outside
isakmp policy 8 authentication pre-share
isakmp policy 8 encryption des
isakmp policy 8 hash md5
isakmp policy 8 group 1
isakmp policy 8 lifetime 86400
```

The **aaa-server** command is used with the **crypto map** command to establish an authentication association so that VPN Clients are authenticated when they access the PIX Firewall.

Refer to the *IPSec User Guide for the Cisco Secure PIX Firewall Version 5.2* for a description of the **crypto** and **isakmp** commands.

# access-group

Binds the access list to an interface. (Configuration mode.)

**access-group** *acl\_ID* **in interface** *interface\_name*

**clear access-group**

**no access-group** *acl\_ID* **in interface** *interface\_name*

**show access-group** *acl\_ID* **in interface** *interface\_name*

## Syntax Description

<i>acl_ID</i>	The name associated with a given access list.
<b>in interface</b>	Filter on inbound packets at the given interface.
<i>interface_name</i>	The name of the network interface.

## Usage Guidelines

The **access-group** command binds an access list to an interface. The access list is applied to traffic inbound to an interface. If you enter the **permit** option in an **access-list** command statement, the PIX Firewall continues to process the packet. If you enter the **deny** option in an **access-list** command statement, PIX Firewall discards the packet and generates the following syslog message:

```
%PIX-4-106019: IP packet from source_addr to destination_addr, protocol protocol received
from interface interface_name deny by access-group acl_ID
```

Always use the **access-list** command with the **access-group** command.



Note

---

The use of **access-group** command overrides the **conduit** and **outbound** command statements for the specified *interface\_name*.

---

The **no access-group** command unbinds the *acl\_ID* from the interface *interface\_name*.

The **show access-group** command displays the current access list bound to the interfaces.

The **clear access-group** command removes all entries from an access list indexed by *acl\_ID*. If *acl\_ID* is not specified, all **access-list** command statements are removed from the configuration.

## Examples

The following example shows use of the **access-group** command:

```
static (inside,outside) 209.165.201.3 10.1.1.3
access-list acl_out permit tcp any host 209.165.201.3 eq 80
access-group acl_out in interface outside
```

The **static** command statement provides a global address of 209.165.201.3 for the web server at 10.1.1.3. The **access-list** command statement lets any host access the global address using port 80. The **access-group** command specifies that the **access-list** command statement applies to traffic entering the outside interface.

# access-list

Create an access list. (Configuration mode.)

```
access-list acl_ID [deny | permit] protocol {source_addr | local_addr} {source_mask | local_mask} operator port {destination_addr | remote_addr} {destination_mask | remote_mask} operator port
```

```
access-list acl_ID [deny | permit] icmp {source_addr | local_addr} {source_mask | local_mask} operator port {destination_addr | remote_addr} {destination_mask | remote_mask} operator port icmp_type
```

```
no access-list acl_ID [[deny | permit] protocol {source_addr | local_addr} {source_mask | local_mask} operator port {destination_addr | remote_addr} {destination_mask | remote_mask} operator port]
```

```
clear access-list [acl_ID [deny | permit] icmp {source_addr | local_addr} {source_mask | local_mask} operator port {destination_addr | remote_addr} {destination_mask | remote_mask} operator port icmp_type]
```

**show access-list**

## Syntax Description

<i>acl_ID</i>	Name of an access list. You can use either a name or number.
<b>deny</b>	<p>When used with the <b>access-group</b> command, the <b>deny</b> option does not allow a packet to traverse the PIX Firewall. By default, PIX Firewall denies all inbound or outbound packets unless you specifically permit access.</p> <p>When used with a <b>crypto map</b> command statement, <b>deny</b> does not select a packet for IPsec protection. The <b>deny</b> option prevents traffic from being protected by IPsec in the context of that particular crypto map entry. In other words, it does not allow the policy as specified in the <b>crypto map</b> command statements to be applied to this traffic.</p>
<b>permit</b>	<p>When used with the <b>access-group</b> command, the <b>permit</b> option selects a packet to traverse the PIX Firewall. By default, PIX Firewall denies all inbound or outbound packets unless you specifically permit access.</p> <p>When used with a <b>crypto map</b> command statement, <b>permit</b> selects a packet for IPsec protection. The <b>permit</b> option causes all IP traffic that matches the specified conditions to be protected by IPsec using the policy described by the corresponding <b>crypto map</b> command statements.</p>
<i>protocol</i>	Name or number of an IP protocol. It can be one of the keywords <b>icmp</b> , <b>ip</b> , <b>tcp</b> , or <b>udp</b> , or an integer in the range 1 to 254 representing an IP protocol number. To match any Internet protocol, including ICMP, TCP, and UDP, use the keyword <b>ip</b> .
<i>source_addr</i>	Address of the network or host from which the packet is being sent. Use this field when an <b>access-list</b> command statement is used in conjunction with an <b>access-group</b> command statement, or with the <b>aaa match access-list</b> command and the <b>aaa authorization</b> command.

<i>source_mask</i>	Netmask bits (mask) to be applied to <i>source_addr</i> , if the source address is for a network mask.
<i>local_addr</i>	Address of the network or host local to the PIX Firewall. Specify a <i>local_addr</i> when the <b>access-list</b> command statement is used in conjunction with a <b>crypto access-list</b> command statement, a <b>nat 0 access-list</b> command statement, or a <b>vpngroup split-tunnel</b> command statement. The <i>local_addr</i> is the address after NAT has been performed.
<i>local_mask</i>	Netmask bits (mask) to be applied to <i>local_addr</i> , if the local address is a network mask.
<i>destination_addr</i>	IP address of the network or host to which the packet is being sent. Specify a <i>destination_addr</i> when the <b>access-list</b> command statement is used in conjunction with an <b>access-group</b> command statement, or with the <b>aaa match access-list</b> command and the <b>aaa authorization</b> command. For inbound connections, <i>destination_addr</i> is the address after NAT has been performed. For outbound connections, <i>destination_addr</i> is the address before NAT has been performed.
<i>destination_mask</i>	Netmask bits (mask) to be applied to <i>destination_addr</i> , if the destination address is a network mask.
<i>remote_addr</i>	IP address of the network or host remote to the PIX Firewall. specify a <i>remote_addr</i> when the <b>access-list</b> command statement is used in conjunction with a <b>crypto access-list</b> command statement, a <b>nat 0 access-list</b> command statement, or a <b>vpngroup split-tunnel</b> command statement.
<i>remote_mask</i>	Netmask bits (mask) to be applied to <i>remote_addr</i> , if the remote address is a network mask
<i>operator</i>	<p>A comparison operand that lets you specify a port or a port range. Use without an operator and port to indicate all ports; for example:</p> <pre>access-list acl_out permit tcp any host 209.165.201.1</pre> <p>Use <b>eq</b> and a port to permit or deny access to just that port. For example, use <b>eq ftp</b> to permit or deny access only to FTP:</p> <pre>access-list acl_out deny tcp any host 209.165.201.1 eq ftp</pre> <p>Use <b>lt</b> and a port to permit or deny access to all ports less than the port you specify. For example, use <b>lt 2025</b> to permit or deny access to the well known ports (1 to 1024):</p> <pre>access-list acl_dmz1 permit tcp any host 192.168.1.1 lt 1025</pre> <p>Use <b>gt</b> and a port to permit or deny access to all ports greater than the port you specify. For example, use <b>gt 42</b> to permit or deny ports 43 to 65535:</p> <pre>access-list acl_dmz1 deny udp any host 192.168.1.2 gt 42</pre> <p>Use <b>neq</b> and a port to permit or deny access to every port except the ports that you specify. For example, use <b>neq 10</b> to permit or deny ports 1-9 and 11 to 65535:</p> <pre>access-list acl_dmz1 deny tcp any host 192.168.1.3 neq 10</pre>

<i>operator</i> (continued)	Use <b>range</b> and a port range to permit or deny access to only those ports named in the range. For example, use <b>range 10 1024</b> to permit or deny access only to ports 10 through 1024. All other ports are unaffected. The use of port ranges can dramatically increase the number of IPSec tunnels. For example, if a port range of 5000 to 65535 is specified for a highly dynamic protocol, up to 60,535 tunnels can be created.  <pre>access-list acl_dmz1 deny tcp any host 192.168.1.4 range ftp telnet</pre>
<i>port</i>	Services you permit or deny access to. Specify services by the port that handles it, such as <b>smtp for port 25</b> , <b>www</b> for port 80, and so on. You can specify ports by either a literal name or a number in the range of 0 to 65535.  You can view valid port numbers online at the following site: <a href="http://www.isi.edu/in-notes/iana/assignments/port-numbers">http://www.isi.edu/in-notes/iana/assignments/port-numbers</a>  See “Ports” in Chapter 1, “Introduction,” for a list of valid port literal names in port ranges; for example, <b>ftp h323</b> . You can also specify numbers.
<i>icmp_type</i>	[Non-IPSec use only]—Permit or deny access to ICMP message types. Refer to Table 5-1 for a list of message types. Omit this option to mean all ICMP types.  ICMP message types are not supported for use with IPSec; that is when the <b>access-list</b> command is used in conjunction with the <b>crypto map</b> command, the <i>icmp_type</i> is ignored.

### Usage Guidelines

The **access-list** command lets you specify if an IP address is permitted or denied access to a port or protocol. In this document, one or more **access-list** command statements with the same access list name are referred to as an “access list.” Access lists associated with IPSec are known as “crypto access lists.” By default, all access in an access list is denied. You must explicitly permit it.

Use the following guidelines for specifying a source, local, or destination address:

- Use a 32-bit quantity in four-part, dotted-decimal format.
- Use the keyword **any** as an abbreviation for an address and mask of 0.0.0.0 0.0.0.0. This keyword is normally not recommended for use with IPSec.
- Use **host address** as an abbreviation for a mask of 255.255.255.255.

Use the following guidelines for specifying a network mask:

- Do not specify a mask if the address is for a host; if the destination address is for a host, use the **host** parameter before the address; for example:

```
access-list acl_grp permit tcp any host 192.168.1.1
```

- If the address is a network address, specify the mask as a 32-bit quantity in four-part, dotted-decimal format. Place zeros in the bit positions you want to ignore.
- Remember that you specify a network mask differently than with the Cisco IOS software **access-list** command. With PIX Firewall, use 255.0.0.0 for a Class A address, 255.255.0.0 for a Class B address, and 255.255.255.0 for a Class C address. If you are using a subnetted network address, use the appropriate network mask; for example:

```
access-list acl_grp permit tcp any 209.165.201.0 255.255.255.224
```

If appropriate, after you have defined an access list, bind it to an interface using the **access-group** command. For IPsec use, bind it with a **crypto map** command statement. In addition, you can bind an access list with the RADIUS authorization feature (described in the next section). Refer to the *IPsec User Guide for the Cisco Secure PIX Firewall Version 5.2* for a description of the **crypto** command.

The **show access-list** command lists the **access-list** command statements in the configuration. The **show access-list** command also lists a hit count that indicates the number of times an element has been matched during an **access-list** command search. The **clear access-list** command removes all **access-list** command statements from the configuration.

The **no access-list** command removes an **access-list** command from the configuration. If you remove all the **access-list** command statements in an access list group, the **no access-list** command also removes the corresponding **access-group** command from the configuration.



Note

---

The **aaa**, **crypto map**, and **icmp** commands make use of the **access-list** command statements.

---

### RADIUS Authorization Feature

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

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

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

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

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

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

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



Note

---

An access list used for RADIUS authorization does not require an **access-group** command to bind the statements to an interface.

---

There is *not* a **radius** option to the **aaa authorization** command.

Follow these steps to enable RADIUS authorization:

- 
- Step 1** Enable RADIUS authentication with the **aaa authentication** command.
  - Step 2** Create the **access-list** command statements to specify what services hosts are authorized to use with RADIUS.
  - Step 3** Configure the authentication server with the vendor-specific **acl=acl\_ID** identifier to specify the **access-list ID**.

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

---

### Usage Notes

1. The **clear access-list** command automatically unbinds an access list from a **crypto map** command or interface. The unbinding of an access list from a **crypto map** command can lead to a condition that discards all packets because the **crypto map** command statements referencing the access list are incomplete. To correct the condition, either define other **access-list** command statements to complete the **crypto map** command statements or remove the **crypto map** command statements that pertain to the **access-list** command statement. Refer to the *IPSec User Guide for the Cisco Secure PIX Firewall Version 5.2* for a description of the **crypto map** command.
2. The **access-list** command operates on a first match basis.
3. If you specify an **access-list** command statement and bind it to an interface with the **access-group** command statement, by default, all traffic inbound to that interface is denied. You must explicitly permit traffic. Note that “inbound” in this context means traffic passing through the interface, rather than the more typical PIX Firewall usage of inbound meaning traffic passing from a lower security level interface to a higher security level interface.
4. Always permit access first and then deny access afterward. If the host entries match, then use a **permit** statement, otherwise use the default **deny** statement. You only need to specify additional **deny** statements if you need to deny specific hosts and permit everyone else.
5. You can view security levels for interfaces with the **show nameif** command.
6. The ICMP message type (*icmp\_type*) option is ignored in IPSec applications because the message type cannot be negotiated with ISAKMP.
7. Only one access list can be bound to an interface using the **access-group** command.
8. If you specify the **permit** option in the access list, the PIX Firewall continues to process the packet. If you specify the **deny** option in the access list, PIX Firewall discards the packet and generates the following syslog message:
 

```
%PIX-4-106019: IP packet from source_addr to destination_addr, protocol protocol
received from interface interface_name deny by access-group acl_ID
```
9. The **access-list** command uses the same syntax as the Cisco IOS software **access-list** command *except* that PIX Firewall uses a subnet mask, whereas Cisco IOS software uses a wildcard mask. (In Cisco IOS software, the mask in this example would be specified with the `0.0.0.255` value.) For example, in the Cisco IOS software **access-list** command, a subnet mask of 0.0.0.255 would be specified as 255.0.0.0 in the PIX Firewall **access-list** command.
10. Cisco recommends that you do not use the **access-list** command with the **conduit** and **outbound** commands. While using these commands together will work, the way in which these commands operate may cause debugging issues because the **conduit** and **outbound** commands operate from

one interface to another whereas the **access-list** command used with the **access-group** command applies only to a single interface. If these commands must be used together, PIX Firewall evaluates the **access-list** command before checking the **conduit** and **outbound** commands.

11. Refer to “Step 13—Add Inbound Server Access” and “Step 14—Add Outbound Access Lists” in Chapter 2, “Configuring the PIX Firewall,” for a detailed description about using the **access-list** command to provide server access and to restrict outbound user access.

### ICMP Message Types

[Non-IPSec use only]—If you prefer more selective ICMP access, you can specify a single ICMP message type as the last option in this command. Table 5-1 lists possible ICMP types values.

**Table 5-1 ICMP Type Literals**

ICMP Type	Literal
0	echo-reply
3	unreachable
4	source-quench
5	redirect
6	alternate-address
8	echo
9	router-advertisement
10	router-solicitation
11	time-exceeded
12	parameter-problem
13	timestamp-reply
14	timestamp-request
15	information-request
16	information-reply
17	mask-request
18	mask-reply
31	conversion-error
32	mobile-redirect

If you specify an ICMP message type for use with IPSec, PIX Firewall ignores it. For example:

```
access-list 10 permit icmp any any echo-reply
```

And IPSec is enabled such that a **crypto map** command references the *acl\_name* for this **access-list** command, then the **echo-reply** ICMP message type is ignored.

### Using the access-list Command with IPSec

If an access list is bound to an interface with the **access-group** command, the access list selects which traffic can traverse the PIX Firewall. When bound to a **crypto map** command statement, the access list selects which IP traffic IPSec protects and which traffic IPSec does not protect. For example, access

lists can be created to protect all IP traffic between Subnet X and Subnet Y or traffic between Host A and Host B. Refer to the *IPSec User Guide for the Cisco Secure PIX Firewall Version 5.2* for a description of the **crypto** command.

The access lists themselves are not specific to IPSec. It is the **crypto map** command statement referencing the specific access list that defines whether IPSec processing is applied to the traffic matching a permit in the access list.

Crypto access lists associated with the IPSec **crypto map** command statement have these primary functions:

- Select outbound traffic to be protected by IPSec (permit = protect).
- Indicate the data flow to be protected by the new security associations (specified by a single permit entry) when initiating negotiations for IPSec security associations.
- Process inbound traffic to filter out and discard traffic that IPSec protects.
- Determine whether or not to accept requests for IPSec security associations on behalf of the requested data flows when processing IKE negotiation from the IPSec peer. (Negotiation is only done for **crypto map** command statements with the **ipsec-isakmp** option.) For a peer's initiated IPSec negotiation to be accepted, it must specify a data flow that is permitted by a crypto access list associated with an **ipsec-isakmp** crypto map entry.

You can associate a crypto access list with an interface by defining the corresponding **crypto map** command statement and applying the crypto map set to an interface. Different access lists must be used in different entries of the same crypto map set. However, both inbound and outbound traffic will be evaluated against the same "outbound" IPSec access list. Therefore, the access list's criteria are applied in the forward direction to traffic exiting your PIX Firewall and the reverse direction to traffic entering your PIX Firewall.

If you want certain traffic to receive one combination of IPSec protection (for example, authentication only) and other traffic to receive a different combination of IPSec protection (for example, both authentication and encryption), you need to create two different crypto access lists to define the two different types of traffic. These different access lists are then used in different crypto map entries that specify different IPSec policies.

Cisco recommends that you configure "mirror image" crypto access lists for use by IPSec and that you avoid using the **any** keyword. See the *IPSec User Guide for the Cisco Secure PIX Firewall Version 5.2* for more information.

If you configure multiple statements for a given crypto access list, in general, the first **permit** statement matched, will be the statement used to determine the scope of the IPSec security association. That is, the IPSec security association will be set up to protect traffic that meets the criteria of the matched statement only. Later, if traffic matches a different **permit** statement of the crypto access list, a new, separate IPSec security association will be negotiated to protect traffic matching the newly matched access list statement.

Some services such as FTP require two **access-list** command statements, one for port 10 and another for port 21, to properly encrypt FTP traffic.

### Examples

The following example creates a numbered access list that specifies a Class C subnet for the source and a Class C subnet for the destination of IP packets. Because the **access-list** command is referenced in the **crypto map** command statement, PIX Firewall encrypts all IP traffic that is exchanged between the source and destination subnets.

```
access-list 101 permit ip 172.21.3.0 255.255.0.0 172.22.2.0 255.255.0.0
access-group 101 in interface outside
crypto map mymap 10 match address 101
[other crypto map commands]
```

The next example only lets an ICMP message type of echo-reply be permitted into the outside interface:

```
access-list acl_out permit icmp any any echo-reply
access-group acl_out interface outside
```

# alias

Administer overlapping addresses with dual NAT. (Configuration mode.)

**alias** [(if\_name)] dnat\_ip foreign\_ip [netmask]

**no alias** [(if\_name)] dnat\_ip foreign\_ip [netmask]

**show alias**

**clear alias**

## Syntax Description

<i>if_name</i>	The internal network interface name in which the <i>foreign_ip</i> overlaps.
<i>dnat_ip</i>	An IP address on the internal network that provides an alternate IP address for the external address that is the same as an address on the internal network.
<i>foreign_ip</i>	IP address on the external network that has the same address as a host on the internal network.
<i>netmask</i>	Network mask applied to both IP addresses. Use 255.255.255.255 for host masks.

## Usage Guidelines

The **alias** command translates one address into another. Use this command to prevent conflicts when you have IP addresses on a network that are the same as those on the Internet or another intranet. You can also use this command to do address translation on a destination address. For example, if a host sends a packet to 209.165.201.1, you can use the **alias** command to redirect traffic to another address, such as, 209.165.201.30.



### Note

You can use the **sysopt nodnsalias** command to disable inbound embedded DNS A record fixups according to aliases that apply to the A record address and outbound replies.



### Note

If the **alias** command is used with the **sysopt ipsec pl-compatible** command, a static **route** command statement must be added for each IP address specified in the **alias** command statement.

After changing or removing an **alias** command statement, use the **clear xlate** command.

There must be an A (address) record in the DNS zone file for the “dnat” address in the **alias** command.

The **alias** command has two uses which can be summarized in the following ways of reading an **alias** command statement:

- If the PIX Firewall gets a packet destined for the *dnat\_IP\_address*, send it to the *foreign\_IP\_address*.
- If the PIX Firewall gets a DNS packet returned to the PIX Firewall destined for *foreign\_network\_address*, alter the DNS packet to change the foreign network address to *dnat\_network\_address*.

The **no alias** command disables a previously set **alias** command statement. The **show alias** command displays **alias** command statements in the configuration. The **clear alias** command removed all **alias** commands from the configuration.

The **alias** command automatically interacts with DNS servers on your network to ensure that domain name access to the aliased IP address is handled transparently.

You can specify a net alias by using network addresses for the *foreign\_ip* and *dnat\_ip* IP addresses. For example, **alias 192.168.201.0 209.165.201.0 255.255.255.224** creates aliases for each IP address between 209.165.201.1 and 209.165.201.30.

**Note**

ActiveX blocking does not occur when users access an IP address referenced by the **alias** command. ActiveX blocking is set with the **filter activex** command.

**Usage Notes**

To access an **alias** *dnat\_ip* address with **static** and **access-list** command statements, specify the *dnat\_ip* address in the **access-list** command statement as the address from which traffic is permitted from. The following example illustrates this note:

```
alias (inside) 192.168.201.1 209.165.201.1 255.255.255.255
static (inside,outside) 209.165.201.1 192.168.201.1 netmask 255.255.255.255
access-list acl_out permit tcp host 192.168.201.1 host 209.165.201.1 eq ftp-data
access-group acl_out in interface outside
```

An alias is specified with the inside address 192.168.201.1 mapping to the foreign address 209.165.201.1.

**Examples**

1. In this example, the inside network contains the IP address 209.165.201.29, which on the Internet belongs to example.com. When inside clients try to access example.com, the packets do not go to the firewall because the client thinks 209.165.201.29 is on the local inside network. To correct this, a net **alias** is created as follows with the **alias** command:

```
alias (inside) 192.168.201.0 209.165.201.0 255.255.255.224
```

```
show alias
```

```
alias 192.168.201.0 209.165.201.0 255.255.255.224
```

When the inside network client 209.165.201.2 connects to example.com, the DNS response from an external DNS server to the internal client's query would be altered by the PIX Firewall to be 192.168.201.29. If the PIX Firewall uses 209.165.200.225 through 209.165.200.254 as the global pool IP addresses, the packet goes to the PIX Firewall with SRC=209.165.201.2 and DST=209.165.201.29. The PIX Firewall translates the address to SRC=209.165.200.254 and DST=209.165.201.29 on the outside.

2. In the next example, a web server is on the inside at 10.1.1.11 and a static for it at 209.165.201.11. The source host is on the outside with address 209.165.201.7. A DNS server on the outside has a record for www.example.com as follows:

```
www.example.com.      IN      A      209.165.201.11
```

The period at the end of the www.example.com. domain name must be included.

The **alias** command follows:

```
alias 10.1.1.11 209.165.201.11 255.255.255.255
```

PIX Firewall doctors the nameserver replies to 10.1.1.11 for inside clients to directly connect to the web server.

The static command statement is as follows:

```
static (inside,outside) 209.165.201.11 10.1.1.11
```

The **access-list** command statement you would expect to use follows:

```
access-list acl_grp permit tcp host 209.165.201.7 host 209.165.201.11 eq telnet
```

But with the **alias** command, use this command:

```
access-list acl_grp permit tcp host 209.165.201.11 eq telnet host 209.165.201.7
```

You can test the DNS entry for the host with the following UNIX **nslookup** command:

```
nslookup -type=any www.example.com
```

# arp

Change or view the ARP cache, and set the timeout value. (Configuration mode.)

```

arp if_name ip_address mac_address [alias]

clear arp

no arp if_name ip_address

show arp [if_name] [ip_address mac_address alias]

arp timeout seconds

no arp timeout

show arp timeout

```

## Syntax Description

<i>if_name</i>	The internal or external interface name specified by the <b>nameif</b> command.
<i>ip_address</i>	Host IP address for the ARP table entry.
<i>mac_address</i>	Hardware MAC address for the ARP table entry; for example, 00e0.1e4e.3d8b.
<b>alias</b>	Make this entry permanent. Alias entries do not time out and are automatically stored in the configuration when you use the <b>write</b> command to store the configuration.
<i>seconds</i>	Duration that an ARP entry can exist in the ARP table before being cleared.

## Usage Guidelines

The **arp** command adds an entry to the PIX Firewall ARP cache. ARP is a low-level TCP/IP protocol that resolves a node's physical address from its IP address through an ARP request asking the node with a particular IP address to send back its physical address. The presence of entries in the ARP cache indicates that the PIX Firewall has network connectivity. The **clear arp** command clears the ARP table but not the **alias** (permanent) entries. Use the **no arp** command to remove these entries. The **show arp** command lists the entries in the ARP table.



### Note

You can use the **sysopt noproxyarp** command to disable proxy-arps on an interface.

Use the **arp** command to add an entry for new hosts you add on your network or when you swap an existing host for another. Alternatively, you can wait for the duration specified with the **arp timeout** command to expire and the ARP table rebuilds itself automatically with the new host information.

The **arp timeout** command sets the duration that an ARP entry can stay in the PIX Firewall ARP table before expiring. The timer is known as the ARP persistence timer. The default value is 14,400 seconds (4 hours).

The **no arp timeout** command sets the timer to its default value. The **show arp timeout** command displays its current value.

### Examples

The following examples illustrate use of the **arp** and **arp timeout** commands:

```
arp inside 192.168.0.42 00e0.1e4e.2a7c
arp outside 192.168.0.43 00e0.1e4e.3d8b alias
show arp
    outside 192.168.0.43 00e0.1e4e.3d8b alias
    inside 192.168.0.42 00e0.1e4e.2a7c
```

```
clear arp inside 192.168.0.42
```

```
arp timeout 42
show arp timeout
arp timeout 42 seconds
```

```
no arp timeout
show arp timeout
arp timeout 14400 seconds
```

# auth-prompt

Change the AAA challenge text. (Configuration mode.)

```
auth-prompt [accept | reject | prompt] string
no auth-prompt [accept | reject | prompt] string
clear auth-prompt
show auth-prompt
```

## Syntax Description

<b>accept</b>	If a user authentication via Telnet is accepted, display the prompt <i>string</i> .
<b>reject</b>	If a user authentication via Telnet is rejected, display the prompt <i>string</i> .
<b>prompt</b>	The AAA challenge prompt string follows this keyword. This keyword is optional for backward compatibility.
<i>string</i>	A string of up to 235 alphanumeric characters. Special characters should not be used; however, spaces and punctuation characters are permitted. Entering a question mark or pressing the <b>Enter</b> key ends the string. (The question mark appears in the string.)

## Usage Guidelines

The **auth-prompt** command lets you change the AAA challenge text for HTTP, FTP, and Telnet access. This text displays above the username and password prompts that users view when logging in. If you do not use this command, FTP users view `FTP authentication`, HTTP users view `HTTP Authentication`, and challenge text does not appear for Telnet access.

If the user authentication occurs from Telnet, you can use the **accept** and **reject** options to display different authentication prompts if the authentication attempt is accepted or rejected by the authentication server.



### Note

---

Microsoft Internet Explorer only displays up to 37 characters in an authentication prompt. Netscape Navigator displays up to 120 characters, and Telnet and FTP display up to 235 characters in an authentication prompt.

---

## Examples

The following example shows how to set the authentication prompt and how users view the prompt:

```
auth-prompt XYZ Company Firewall Access
```

After this string is added to the configuration, users view:

```
XYZ Company Firewall Access
User Name:
Password:
```

The **prompt** keyword can be included or omitted. For example:

```
auth-prompt prompt Hello There!
```

This command statement is the same as the following:

```
auth-prompt Hello There!
```

# clear Commands

Remove commands from the configuration or reset command values (All modes.)

Table 5-2, Table 5-3, and Table 5-4 list each mode in which the **clear** commands first appear. Each **clear** command listed in one mode can be also accessed in each subsequent more secure mode going from unprivileged to configuration mode, but not from less secure modes.


**Note**

For IPsec **clear** commands, refer to the *IPSec User Guide for the Cisco Secure PIX Firewall Version 5.2*.

**Table 5-2 Unprivileged Mode Clear Commands**

Clear Command	Description	Described on Command Page
<b>clear pager</b>	Resets the number of displayed lines to 24.	<b>pager</b>

**Table 5-3 Privileged Mode Clear Commands**

Clear Command	Description	Described on Command Page
<b>clear arp</b>	Clears the ARP table.	<b>arp</b>
<b>clear auth-prompt</b>	Removes an <b>auth-prompt</b> command statement from the configuration.	<b>auth-prompt</b>
<b>clear blocks</b>	Resets the <b>show blocks</b> command statement counters.	<b>show blocks/clear blocks</b>
<b>clear configure</b>	Resets command parameters in the configuration to their default values.	<b>configure</b>
<b>clear flashfs</b>	Clears Flash memory prior to downgrading the PIX Firewall software version.	<b>flashfs</b>
<b>clear local-host</b>	Resets the information displayed for the <b>show local-host</b> command.	<b>local-host (clear and show)</b>
<b>clear passwd</b>	Resets the Telnet password back to “cisco.”	<b>passwd</b>
<b>clear traffic</b>	Resets the counters for the show traffic command.	<b>show traffic/clear traffic</b>
<b>clear uauth</b>	Deletes one user’s or all users’ AAA authorization caches, which forces the user or users to reauthenticate the next time they create a connection.	<b>uauth (clear and show)</b>
<b>clear xlate</b>	Clears the contents of the translation slots.	<b>xlate (clear and show)</b>

Table 5-4 Configuration Mode Clear Commands

Clear Command	Description	Described on Command Page
<b>clear aaa</b>	Remove <b>aaa</b> command statements from the configuration.	<b>aaa</b>
<b>clear access-list</b>	Remove <b>access-list</b> command statements from the configuration. This command also stops all traffic through the PIX Firewall on the affected <b>access-list</b> command statements.	<b>access-list</b>
<b>clear access-group</b>	Removes <b>access-group</b> command statements from the configuration.	<b>access-group</b>
<b>clear alias</b>	Removes <b>alias</b> command statements from the configuration.	<b>alias</b>
<b>clear apply</b>	Removes <b>apply</b> command statements from the configuration.	<b>outbound/apply</b>
<b>clear conduit</b>	Removes <b>conduit</b> command statements from the configuration.	<b>conduit</b>
<b>clear dhcpd</b>	Removes <b>dhcpd</b> command statements from the configuration.	<b>dhcpd</b>
<b>clear established</b>	Removes <b>established</b> command statements from the configuration.	<b>established</b>
<b>clear filter</b>	Removes <b>filter</b> command statements from the configuration.	<b>filter</b>
<b>clear fixup</b>	Resets <b>fixup protocol</b> command statements to their default values.	<b>fixup protocol</b>
<b>clear flashfs</b>	Clears Flash memory before downgrading to a previous PIX Firewall version.	<b>flashfs</b>
<b>clear global</b>	Removes <b>global</b> command statements from the configuration.	<b>global</b>
<b>clear icmp</b>	Removes <b>icmp</b> command statements from the configuration.	<b>icmp</b>
<b>clear ip</b>	Sets all PIX Firewall interface IP addresses to 127.0.0.1 and stops all traffic.	<b>ip address</b>
<b>clear interface</b>	Clear counters for the <b>show interface</b> command.	<b>interface</b>
<b>clear logging</b>	Clear syslog message queue accumulated by the <b>logging buffered</b> command.	<b>logging</b>
<b>clear names</b>	Removes <b>name</b> command statements from the configuration.	<b>name/names</b>
<b>clear nameif</b>	Reverts <b>nameif</b> command statements to default interface names and security levels.	<b>nameif</b>
<b>clear nat</b>	Removes <b>nat</b> command statements from the configuration.	<b>nat</b>
<b>clear outbound</b>	Removes <b>outbound</b> command statements from the configuration.	<b>outbound/apply</b>

Table 5-4 Configuration Mode Clear Commands (continued)

Clear Command	Description	Described on Command Page
<b>clear rip</b>	Removes <b>rip</b> command statements from the configuration.	<b>rip</b>
<b>clear route</b>	Removes <b>route</b> command statements from the configuration that do not contain the <b>CONNECT</b> keyword.	<b>route</b>
<b>clear snmp-server</b>	Removes <b>snmp-server</b> command statements from the configuration.	<b>snmp-server</b>
<b>clear ssh</b>	Removes <b>ssh</b> command statement from the configuration.	<b>ssh</b>
<b>clear static</b>	Removes <b>static</b> command statements from the configuration.	<b>static</b>
<b>clear sysopt</b>	Removes <b>sysopt</b> command statements from the configuration.	<b>sysopt</b>
<b>clear telnet</b>	Removes <b>telnet</b> command statements from the configuration.	<b>telnet</b>
<b>clear tftp-server</b>	Removes <b>tftp-server</b> command statements from the configuration.	<b>tftp-server</b>
<b>clear timeout</b>	Resets <b>timeout</b> command durations to their default values.	<b>timeout</b>
<b>clear url-cache</b>	Removes <b>url-cache</b> command statements from the configuration.	<b>url-cache</b>
<b>clear url-server</b>	Removes <b>url-server</b> command statements from the configuration.	<b>url-server</b>
<b>clear virtual</b>	Removes <b>virtual</b> command statements from the configuration.	<b>virtual</b>
<b>clear vpdn</b>	Removes <b>vpdn</b> command statements from the configuration.	<b>vpdn</b>

# clock

Set the PIX Firewall clock for use with the PIX Firewall Syslog Server and the Public Key Infrastructure (PKI) protocol. (Configuration mode.)

**clock**

**clock set** *hh:mm:ss month day year*

**clock set** *hh:mm:ss day month year*

**show clock**

## Syntax Description

<i>hh:mm:ss</i>	The current hour:minutes:seconds expressed in 24-hour time; for example, <b>20:54:00</b> for 8:54 pm. Zeros can be entered as a single digit; for example, <b>21:0:0</b> .
<i>month</i>	The current month expressed as the first three characters of the month; for example, <b>apr</b> for April.
<i>day</i>	The current day of the month; for example, <b>1</b> .
<i>year</i>	The current year expressed as four digits; for example, <b>2000</b> .

## Usage Guidelines

The **clock** command lets you specify the current time, month, day, and year for use time stamped syslog messages, which you can enable with the **logging timestamp** command. You can view the current time with the **clock** or the **show clock** command.



### Note

---

The lifetime of a certificate and the Certificate Revocation List (CRL) is checked in GMT. If you are using IPSec with certificates, set the PIX Firewall clock to GMT timezone to ensure that CRL checking works correctly.

---

You can interchange the settings for the *day* and the *month*; for example, **clock set 21:0:0 1 apr 2000**.

A time prior to January 1, 1998 or after December 31, 2097 will not be accepted (the maximum date that the **clock** command can work to).

While the PIX Firewall clock is year 2000 compliant, it does not adjust itself for daylight savings time changes; however, it does know about leap years.

The PIX Firewall clock setting is retained in memory when the power is off by a battery on the PIX Firewall unit's motherboard. Should this battery fail, contact Cisco's customer support for a replacement PIX Firewall unit.

Cisco's PKI (Public Key Infrastructure) protocol uses the clock to make sure that a Certificate Revocation List (CRL) is not expired. Otherwise, the CA may reject or allow certificates based on an incorrect timestamp. Refer to the *IPSec User Guide for the Cisco Secure PIX Firewall Version 5.2* for a description of IPSec concepts.

### Examples

To enable PFSS time-stamp logging for the first time, use these commands:

```
clock set 21:0:0 apr 1 2000
show clock
21:00:05 Apr 01 2000
logging host 209.165.201.3
logging timestamp
logging trap 5
```

In this example, the **clock** command sets the clock to 9 pm on April 1, 2000. The **logging host** command specifies that a syslog server is at IP address 209.165.201.3. The PIX Firewall automatically determines that the server is a PFSS and sends syslog messages to it via TCP and UDP. The **logging timestamp** command enables sending time stamped syslog messages. The **logging trap 5** command in this example specifies that messages at syslog level 0 through 5 be sent to the syslog server. The value 5 is used to capture severe and normal messages, but also those of the **aaa authentication enable** command.

# conduit

Add, delete, or show conduits through the PIX Firewall for incoming connections. (Configuration mode.)

**conduit permit** | **deny** *protocol global\_ip global\_mask [operator port [port]] foreign\_ip foreign\_mask [operator port [port]]*

**no conduit permit** | **deny** *protocol global\_ip global\_mask [operator port [port]] foreign\_ip foreign\_mask [operator port [port]]*

**conduit permit** | **deny icmp** *global\_ip global\_mask foreign\_ip foreign\_mask [icmp\_type]*

**clear conduit**

**show conduit**

## Syntax Description

<b>permit</b>	Permit access if the conditions are matched.
<b>deny</b>	Deny access if the conditions are matched.
<i>protocol</i>	Specify the transport protocol for the connection. Possible literal values are <b>icmp</b> , <b>tcp</b> , <b>udp</b> , or an integer in the range 0 through 255 representing an IP protocol number. Use <b>ip</b> to specify all transport protocols. You can view valid protocol numbers online at the following site:  <a href="http://www.isi.edu/in-notes/iana/assignments/protocol-numbers">http://www.isi.edu/in-notes/iana/assignments/protocol-numbers</a>  If you specify the icmp protocol, you can permit or deny ICMP access to one or more global IP addresses. Specify the ICMP type in the <i>icmp_type</i> variable, or omit to specify all ICMP types. See the Usage Guidelines for a complete list of the ICMP types.
<i>global_ip</i>	A global IP address previously defined by a <b>global</b> or <b>static</b> command. You can use <b>any</b> if the <i>global_ip</i> and <i>global_mask</i> are 0.0.0.0 0.0.0.0. The <b>any</b> option applies the <b>permit</b> or <b>deny</b> parameters to the global addresses.  If <i>global_ip</i> is a host, you can omit <i>global_mask</i> by specifying the <b>host</b> command before <i>global_ip</i> . For example:  <pre>conduit permit tcp host 209.165.201.1 eq ftp any</pre> This example lets any foreign host access global address 209.165.201.1 for FTP.
<i>global_mask</i>	Network mask of <i>global_ip</i> . The <i>global_mask</i> is a 32-bit, four-part dotted decimal; such as, 255.255.255.255. Use zeros in a part to indicate bit positions to be ignored. Use subnetting if required. If you use <b>0</b> for <i>global_ip</i> , use <b>0</b> for the <i>global_mask</i> ; otherwise, enter the <i>global_mask</i> appropriate to <i>global_ip</i> .

*foreign\_ip* An external IP address (host or network) that can access the *global\_ip*. You can specify **0.0.0.0** or **0** for any host. If both the *foreign\_ip* and *foreign\_mask* are 0.0.0.0 0.0.0.0, you can use the shorthand **any** option.

If *foreign\_ip* is a host, you can omit *foreign\_mask* by specifying the **host** command before *foreign\_ip*. For example:

```
conduit permit tcp any eq ftp host 209.165.201.2
```

This example lets foreign host 209.165.201.2 access any global address for FTP.

*foreign\_mask* Network mask of *foreign\_ip*. The *foreign\_mask* is a 32-bit, four-part dotted decimal; such as, 255.255.255.255. Use zeros in a part to indicate bit positions to be ignored. Use subnetting if required. If you use **0** for *foreign\_ip*, use **0** for the *foreign\_mask*; otherwise, enter the *foreign\_mask* appropriate to *foreign\_ip*. You can also specify a mask for subnetting, for example, 255.255.255.192.

*operator* A comparison operand that lets you specify a port or a port range.

Use without an operator and port to indicate all ports; for example:

```
conduit permit tcp any any
```

Use **eq** and a port to permit or deny access to just that port. For example use **eq ftp** to permit or deny access only to FTP:

```
conduit deny tcp host 192.168.1.1 eq ftp 209.165.201.1
```

Use **lt** and a port to permit or deny access to all ports less than the port you specify. For example, use **lt 2025** to permit or deny access to the well known ports (1 to 1024):

```
conduit permit tcp host 192.168.1.1 lt 1025 any
```

Use **gt** and a port to permit or deny access to all ports greater than the port you specify. For example, use **gt 42** to permit or deny ports 43 to 65535:

```
conduit deny udp host 192.168.1.1 gt 42 host 209.165.201.2
```

Use **neq** and a port to permit or deny access to every port except the ports that you specify.

For example, use **neq 10** to permit or deny ports 1-9 and 11 to 65535:

```
conduit deny tcp host 192.168.1.1 neq 10 host 209.165.201.2 neq 42
```

Use **range** and a port range to permit or deny access to only those ports named in the range.

For example, use **range 10 1024** to permit or deny access only to ports 10 through 1024. All other ports are unaffected.

```
conduit deny tcp any range ftp telnet any
```

By default, all ports are denied until explicitly permitted.

*port* Service(s) you permit to be used while accessing *global\_ip* or *foreign\_ip*. Specify services by the port that handles it, such as **smtp for port 25**, **www** for port 80, and so on. You can specify ports by either a literal name or a number in the range of 0 to 65535. You can specify all ports by not specifying a port value; for example:

```
conduit deny tcp any any
```

This command is the default condition for the **conduit** command in that all ports are denied until explicitly permitted.

You can view valid port numbers online at the following site:

<http://www.isi.edu/in-notes/iana/assignments/port-numbers>

See “Ports” in Chapter 1, “Introduction,” for a list of valid port literal names in port ranges; for example, **ftp h323**. You can also specify numbers.

*icmp\_type* The type of ICMP message. Table 5-5 lists the ICMP type literals that you can use in this command. Omit this option to mean all ICMP types. An example of this command that permits all ICMP types is **conduit permit icmp any any**. This command lets ICMP pass inbound and outbound.

### Usage Guidelines

A **conduit** command statement creates an exception to the PIX Firewall Adaptive Security mechanism by permitting connections from one firewall network interface to access hosts on another.

The **clear conduit** command removes all **conduit** command statements from your configuration.

The **conduit** command can permit or deny access to either the **global** or **static** commands; however, neither is required for the **conduit** command. You can associate a **conduit** command statement with a **global** or **static** command statement through the global address, either specifically to a single global address, a range of global addresses, or to all global addresses.



#### Note

The **conduit** command has been superseded by the **access-list** command. We recommend that you migrate your configuration away from the **conduit** command to maintain future compatibility.

When used with a **static** command statement, a **conduit** command statement permits users on a lower security interface to access a higher security interface. When not used with a **static** command statement, a **conduit** command statement permits both inbound and outbound access.

### Converting conduit Commands to access-list Commands

Follow these steps to convert **conduit** command statements to **access-list** commands:

- Step 1** View the **static** command format. This command normally precedes both the **conduit** and **access-list** commands. The **static** command syntax is as follows:

```
static (high_interface,low_interface) global_ip local_ip netmask mask
```

For example:

```
static (inside,outside) 209.165.201.5 192.168.1.5 netmask 255.255.255.255
```

This command maps the global IP address 209.165.201.5 on the outside interface to the web server 192.168.1.5 on the inside interface. The 255.255.255.255 is used for host addresses.

- Step 2** View the **conduit** command format. The **conduit** command is similar to the **access-list** command in that it restricts access to the mapping provided by the **static** command. The **conduit** command syntax is as follows:

```
conduit action protocol global_ip global_mask global_operator global_port [global_port] foreign_ip
foreign_mask foreign_operator foreign_port [foreign_port]
```

For example:

```
conduit permit tcp host 209.165.201.5 eq www any
```

This command permits TCP for the global IP address 209.165.201.5 that was specified in the **static** command statement and permits access over port 80 (**www**). The “**any**” option lets any host on the outside interface access the global IP address.

The **static** command identifies the interface that the **conduit** command restricts access to.

- Step 3** Create the **access-list** command from the **conduit** command options. The *acl\_name* in the **access-list** command is a name or number you create to associate **access-list** command statements with an **access-group** or **crypto map** command statement.

Normally the **access-list** command format is as follows:

```
access-list acl_name [deny | permit] protocol src_addr src_mask operator port dest_addr dest_mask
operator port
```

However, using the syntax from the **conduit** command in the **access-list** command, you can see how the *foreign\_ip* in the **conduit** command is the same as the *src\_addr* in the **access-list** command and how the *global\_ip* option in the **conduit** command is the same as the *dest\_addr* in the **access-list** command. The **access-list** command syntax overlaid with the **conduit** command options is as follows:

```
access-list acl_name action protocol foreign_ip foreign_mask foreign_operator foreign_port
[foreign_port] global_ip global_mask global_operator global_port [global_port]
```

For example:

```
access-list acl_out permit tcp any host 209.165.201.5 eq www
```

This command identifies the **access-list** command statement group with the “**acl\_out**” identifier. You can use any name or number for your own identifier. (In this example the identifier, “acl” is from ACL, which means Access Control List and “out” is an abbreviation for the outside interface.) It makes your configuration clearer if you use an identifier name that indicates the interface to which you are associating the **access-list** command statements. The example **access-list** command, like the **conduit** command, permits TCP connections from any system on the outside interface. The **access-list** command is associated with the outside interface with the **access-group** command.

- Step 4** Create the **access-group** command using the *acl\_name* from the **access-list** command and the *low\_interface* option from the **static** command. The format for the **access-group** command is as follows:

```
access-group acl_name in interface low_interface
```

For example:

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

This command associates with the “**acl\_out**” group of **access-list** command statements and states that the **access-list** command statement restricts access to the outside interface.

#### More on the conduit Command

If you associate a **conduit** command statement with a **static** command statement, only the interfaces specified on the **static** command statement have access to the **conduit** command statement. For example, if a **static** command statement lets users on the dmz interface access a server on the inside interface, only users on the dmz interface can access the server via the **static** command statement. Users on the outside do not have access.



Note

The **conduit** command statements are processed in the order entered into the configuration.

The **permit** and **deny** options for the **conduit** command are processed in the order listed in the PIX Firewall configuration. In the following example, host 209.165.202.129 is not denied access through the PIX Firewall because the **permit** option precedes the **deny** option:

```
conduit permit tcp host 209.165.201.4 eq 80 any
conduit deny tcp host 209.165.201.4 host 209.165.202.129 eq 80 any
```



Note

If you want internal users to be able to ping external hosts, use the **conduit permit icmp any any** command.

After changing or removing a **conduit** command statement, use the **clear xlate** command.

You can remove a **conduit** command statement with the **no conduit** command. Use the **show conduit** command to view the **conduit** command statements in the configuration and the number of times (hit count) an element has been matched during a **conduit** command search.

If you prefer more selective ICMP access, you can specify a single ICMP message type as the last option in this command. Table 5-5 lists possible ICMP types values.

**Table 5-5 ICMP Type Literals**

ICMP Type	Literal
0	echo-reply
3	unreachable
4	source-quench
5	redirect
6	alternate-address
8	echo
9	router-advertisement
10	router-solicitation
11	time-exceeded

Table 5-5 ICMP Type Literals (continued)

ICMP Type	Literal
12	parameter-problem
13	timestamp-reply
14	timestamp-request
15	information-request
16	information-reply
17	mask-request
18	mask-reply
31	conversion-error
32	mobile-redirect

### Usage Notes

1. By default, all ports are denied until explicitly permitted.
2. The **conduit** command statements are processed in the order entered in the configuration. If you remove a command, it affects the order of all subsequent **conduit** command statements.
3. To remove all **conduit** command statements, cut and paste your configuration onto your console computer, edit the configuration on the computer, use the **write erase** command to clear the current configuration, and then paste the configuration back into the PIX Firewall.
4. If you use PAT (Port Address Translation), you cannot use a **conduit** command statement using the PAT address to either permit or deny access to ports.
5. Two **conduit** command statements are required for establishing access to the following services: **discard**, **dns**, **echo**, **ident**, **pptp**, **rpc**, **sunrpc**, **syslog**, **tacacs-ds**, **talk**, and **time**. Each service, except for **pptp**, requires one **conduit** for TCP and one for UDP. For DNS, if you are only receiving zone updates, you only need a single **conduit** command statement for TCP.

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

```
static (dmz2,outside) 209.165.201.5 192.168.1.5 netmask 255.255.255.255
conduit permit tcp host 209.165.201.5 eq 1723 any
conduit permit gre host 209.165.201.5 any
```

In this example, PPTP is being used to handle access to host 192.168.1.5 on the dmz2 interface from users on the outside. Outside users access the dmz2 host using global address 209.165.201.5. The first **conduit** command statement opens access for the PPTP protocol and gives access to any outside users. The second **conduit** command statement permits access to GRE. If PPTP was not involved and GRE was, you could omit the first **conduit** command statement.

6. The RPC **conduit** command support fixes up UDP portmapper and rpcbind exchanges. TCP exchanges are not supported. This lets simple RPC-based programs work; however, remote procedure calls, arguments, or responses that contain addresses or ports will not be fixed up. For MSRPC, two **conduit** command statements are required, one for port 135 and another for access to the high ports (1024-65535). For Sun RPC, a single **conduit** command statement is required for UDP port 111.

Once you create a **conduit** command statement for RPC, you can use the following command to test its activity from a UNIX host:

```
rpcinfo -u unix_host_ip_address 150001
```

Replace *unix\_host\_ip\_address* with the IP address of the UNIX host.

7. You can overlay host statics on top of a net static range to further refine what an individual host can access:

```
static (inside, outside) 209.165.201.0 10.1.1.0 netmask 255.255.255.0
conduit permit tcp 209.165.201.0 255.255.255.0 eq ftp any
static (inside, outside) 203.31.17.3 10.1.1.3 netmask 255.255.255.0
conduit permit udp host 209.165.201.3 eq h323 host 209.165.202.3
```

In this case, the host at 209.165.202.3 has InternetPhone access in addition to its blanket FTP access.

### Examples

1. The following commands permit access between an outside UNIX gateway host at 209.165.201.2, to an inside SMTP server with Mail Guard at 192.168.1.49. Mail Guard is enabled in the default configuration for PIX Firewall with the **fixup protocol smtp 25** command. The global address on the PIX Firewall is 209.165.201.1:

```
static (inside,outside) 209.165.201.1 192.168.1.49 netmask 255.255.255.255 0 0
conduit permit tcp host 209.165.201.1 eq smtp host 209.165.201.2
```

To disable Mail Guard, enter the following command:

```
no fixup protocol smtp 25
```

2. You can set up an inside host to receive H.323 InternetPhone calls and allow the outside network to connect inbound via the IDENT protocol (TCP port 113). In this example, the inside network is at 192.168.1.0, the global addresses on the outside network are referenced via the 209.165.201.0 network address with a 255.255.255.224 mask:

```
static (inside,outside) 209.165.201.0 192.168.1.0 netmask 255.255.255.224 0 0
conduit permit tcp 209.165.201.0 255.255.255.224 eq h323 any
conduit permit tcp 209.165.201.0 255.255.255.224 eq 113 any
```

3. You can create a web server on the perimeter interface that can be accessed by any outside host as follows:

```
static (perimeter,outside) 209.165.201.4 192.168.1.4 netmask 255.255.255.255 0 0
conduit permit tcp host 209.165.201.4 eq 80 any
```

In this example, the **static** command statement maps the perimeter host, 192.168.1.4, to the global address, 209.165.201.4. The **conduit** command statement specifies that the global host can be accessed on port 80 (web server) by any outside host.

# configure

Clear or merge current configuration with that on floppy or Flash memory, start configuration mode, or view current configuration. (Privileged mode.)



Note

The PIX 506, PIX 515, and PIX 525 do not support use of the **configure floppy** command.

**clear configure primary | secondary | all**

**configure net** [[*server\_ip*]:*filename*]

**configure floppy**

**configure memory**

**configure terminal**

**show configure**

## Syntax Description

<b>clear</b>	Clears aspects of the current configuration in RAM. Use the <b>write erase</b> command to clear the complete configuration.
<b>primary</b>	Sets the <b>interface</b> , <b>ip</b> , <b>mtu</b> , <b>nameif</b> , and <b>route</b> commands to their default values. In addition, interface names are removed from all commands in the configuration.
<b>secondary</b>	Removes the <b>aaa-server</b> , <b>alias</b> , <b>access-list</b> , <b>apply</b> , <b>conduit</b> , <b>global</b> , <b>outbound</b> , <b>static</b> , <b>telnet</b> , and <b>url-server</b> command statements from your configuration.
<b>net</b>	Loads the configuration from a TFTP server and the path you specify.
<b>all</b>	Combines the <b>primary</b> and <b>secondary</b> options.
<b>floppy</b>	Merges the current configuration with that on diskette.
<b>memory</b>	Merges the current configuration with that in Flash memory.
<b>terminal</b>	Starts configuration mode to enter configuration commands from a terminal. Exit configuration mode by entering the <b>quit</b> command.
<i>server_ip</i>	Merges the current configuration with that available across the network at another location, which is defined with the <b>tftp-server</b> command.
<i>filename</i>	A filename you specify to qualify the location of the configuration file on the TFTP server named in <i>server_ip</i> . If you set a filename with the <b>tftp-server</b> command, do not specify it in the configure command; instead just use a colon (:) without a filename.

### Usage Guidelines

The **clear configure** command resets a configuration to its default values. Use this command to create a template configuration or when you want to clear all values. The **clear configure primary** command resets the default values for the **interface**, **ip**, **mtu**, **nameif**, and **route** commands. This command also deletes interface names in the configuration.

The **clear configure secondary** command removes the **aaa-server**, **alias**, **access-list**, **apply**, **conduit**, **global**, **outbound**, **static**, **telnet**, and **url-server** command statements from the configuration. However, the **clear configure secondary** command does not remove **tftp-server** command statements.



#### Note

Save your configuration before using the **clear configure** command. The **clear configure secondary** command does not prompt you before deleting lines from your configuration.

The **configure net** command merges the current running configuration with a TFTP configuration stored at the IP address you specify and from the file you name. If you specify both the IP address and path name in the **tftp-server** command, you can specify *:filename* as simply a colon (:). For example:

```
configure net :
```

Use the **write net** command to store the configuration in the file.

If you have an existing PIX Firewall configuration on a TFTP server and store a shorter configuration with the same file name on the TFTP server, some TFTP servers will leave some of the original configuration after the first “:end” mark. This does not affect the PIX Firewall because the **configure net** command stops reading when it reaches the first “:end” mark. However, this may cause confusion if you view the configuration and see extra text at the end of the configuration. This does not occur if you are using Cisco TFTP Server version 1.1 for Windows NT.



#### Note

Many TFTP servers require the configuration file to be world-readable to be accessible.

The **configure floppy** command merges the current running configuration with the configuration stored on diskette. This command assumes that the diskette was previously created by the **write floppy** command.

The **configure memory** command merges the configuration in Flash memory into the current configuration in RAM.

The **configure terminal** command starts configuration mode. Exit configuration mode with the **quit** command. After exiting configuration mode, use **write memory** to store your changes in Flash memory or **write floppy** to store the configuration on diskette. Use the **write terminal** command to display the current configuration.

The **show configure** command lists the contents of the configuration in Flash memory.

Each command statement from diskette (with **configure floppy**), Flash memory (with **configure memory**), or TFTP transfer (with **configure net**) is read into the current configuration and evaluated in the same way as commands entered from a keyboard with these rules:

- If the command on diskette or Flash memory is identical to an existing command in the current configuration, it is ignored.
- If the command on diskette or Flash memory is an additional instance of an existing command, such as if you already have one **telnet** command for IP address 10.2.3.4 and the diskette configuration has a **telnet** command for 10.7.8.9, then both commands appear in the current configuration.

- If the command redefines an existing command, the command on diskette or Flash memory overwrites the command in the current configuration in RAM. For example, if you have **hostname ram** in the current configuration and **hostname floppy** on diskette, the command in the configuration becomes **hostname floppy** and the command line prompt changes to match the new host name when that command is read from diskette.

### Examples

The following example shows how to configure the PIX Firewall using a configuration retrieved with TFTP:

```
configure net 10.1.1.1:/tftp/config/pixconfig
```

The `pixconfig` file is stored on the TFTP server at 10.1.1.1 in the `tftp/config` folder.

The following example shows how to configure the PIX Firewall from a diskette:

```
configure floppy
```

The following example shows how to configure the PIX Firewall from the configuration stored in Flash memory:

```
configure memory
```

The following example shows the commands you enter to access configuration mode, view the configuration, and save it in Flash memory.

Access privileged mode with the **enable** command and configuration mode with the **configure terminal** command. View the current configuration with the **write terminal** command and save your configuration to Flash memory using the **write memory** command.

```
pixfirewall> enable
password:
pixfirewall# configure terminal
pixfirewall(config)# write terminal
: Saved
... config commands ...
: End
```

```
write memory
```

# copy tftp flash

Change software images without requiring access to the TFTP monitor mode. (Configuration mode.)

```
copy tftp:[[//location]//pathname]] flash
```

## Syntax Description

<b>copy tftp flash</b>	Download Flash memory software images via TFTP without using monitor mode. An image you download is made available to the PIX Firewall on the next reload (reboot).
<i>location</i>	This either an IP address or a name that resolves to an IP address via the PIX Firewall naming resolution mechanism (currently static mappings via the <b>name</b> and <b>names</b> commands).
<i>pathname</i>	PIX Firewall must know how to reach this location via its routing table information. This information is determined by the <b>ip address</b> command, the <b>route</b> command, or also RIP, depending upon your configuration. The <i>pathname</i> can include any directory names besides the actual last component of the path to the file on the server.

## Usage Guidelines

The **copy tftp flash** command lets you download a software image via TFTP. You can use the **copy tftp flash** command with any PIX Firewall model running version 5.1 or later.

The image you download is made available to the PIX Firewall on the next reload (reboot).

The command syntax is as follows:

```
copy tftp:[[//location]//pathname]] flash
```

If the command is used without the *location* or *pathname* optional parameters, then the location and filename are obtained from the user interactively via a series of questions similar to those presented by Cisco IOS software. If you only enter a colon (:), parameters are taken from the **tftp-server** command settings. If other optional parameters are supplied, then these values would be used in place of the corresponding **tftp-server** command setting. Supplying any of the optional parameters, such as a colon and anything after it, causes the command to run without prompting for user input.

The *location* is either an IP address or a name that resolves to an IP address via the PIX Firewall naming resolution mechanism (currently static mappings via the **name** and **names** commands). PIX Firewall must know how to reach this location via its routing table information. This information is determined by the **ip address** command, the **route** command, or also RIP, depending upon your configuration.

The *pathname* can include any directory names besides the actual last component of the path to the file on the server. The pathname cannot contain spaces. If a directory name has spaces, set the directory in the TFTP server instead of in the **copy tftp flash** command.

If your TFTP server has been configured to point to a directory on the system from which you are downloading the image, you need only use the IP address of the system and the image filename.

For example, if you want to download the `pix512.bin` file from the D: partition on a Windows system (IP address 10.1.1.5), you would access the Cisco TFTP Server **View>Options** menu and enter the filename path in the **TFTP server root directory** edit box; for example, `D:\pix_images`. To copy the file to the PIX Firewall, use the following **copy tftp** command:

```
copy tftp://10.1.1.5/pix512.bin flash
```

The TFTP server receives the command and determines the actual file location from its root directory information. The server then downloads the TFTP image to the PIX Firewall.

**Note**


---

Images prior to version 5.1 cannot be retrieved using this mechanism.

---

**Examples**

The following example causes the PIX Firewall to prompt you for the filename and location before you start the TFTP download:

```
copy tftp flash
Address or name of remote host [127.0.0.1]? 10.1.1.5
Source file name [cdisk]? pix512.bin
copying tftp://10.1.1.5/pix512.bin to flash
[yes|no|again]? yes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!...
Received 1695744 bytes.
Erasing current image.
Writing 1597496 bytes of image.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!...
Image installed.
```

The next example takes the information from the **tftp-server** command. In this case, the TFTP server is in an intranet and resides on the outside interface. The example sets the filename and location from the **tftp-server** command, saves memory, and then downloads the image to Flash memory:

```
tftp-server outside 10.1.1.5 pix512.bin
Warning: 'outside' interface has a low security level (0).
write memory
Building configuration...
Cryptochecksum: 017c452b d54be501 8620ba48 490f7e99
[OK]
copy tftp: flash
copying tftp://10.1.1.5/pix512.bin to flash
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!...
```

The next example overrides the information in the **tftp-server** command to let you specify alternate information about the filename and location. If you have not set the **tftp-server** command, you can also use the **copy tftp flash** command to specify all information as shown in the second example that follows:

```
copy tftp://pix512.bin flash
copy tftp://10.0.0.1/pix512.bin flash
```

The next example maps an IP address to the **tftp-host** name with the **name** command and uses the **tftp-host** name in the **copy** commands:

```
name 10.1.1.6 tftp-host
copy tftp://tftp-host/pix512.bin flash
copy tftp://tftp-host/tftpboot/pix512.bin flash
```

# debug

Debug packets or ICMP tracings through the PIX Firewall. (Configuration mode.)

**debug dhcpc detail | error | packet**

**no debug dhcpc detail | error | packet**

**debug dhcpd event | packet**

**no debug dhcpd event | packet**

**debug fover *option***

**no debug fover *option***

**debug h323 h225 [asn | event]**

**no debug h323 h225 [asn | event]**

**debug h323 h245 [asn | event]**

**no debug h323 h245 [asn | event]**

**debug h323 ras [asn | event]**

**no debug h323 ras [asn | event]**

**debug icmp trace**

**no debug icmp trace**

**debug packet *if\_name* [src *source\_ip* [netmask *mask*]] [dst *dest\_ip* [netmask *mask*]]  
[[proto icmp] | [proto tcp [sport *src\_port*] [dport *dest\_port*]] |  
[proto udp [sport *src\_port*] [dport *dest\_port*]] [rx | tx | both]**

**no debug packet *if\_name* [src *source\_ip* [netmask *mask*]] [dst *dest\_ip* [netmask *mask*]]  
[[proto icmp] | [proto tcp [sport *src\_port*] [dport *dest\_port*]] |  
[proto udp [sport *src\_port*] [dport *dest\_port*]] [rx | tx | both]**

**debug ppp error | io / uauth**

**no debug ppp error | io / uauth**

**debug sqlnet**

**no debug sqlnet**

**debug ssh**

**no debug ssh**

**debug vpdn event | error | packet**

**no debug vpdn event | error | packet**

**show debug**

#### Syntax Description

<b>dhcpc detail</b>	Display detailed information about the DHCP client packets.
<b>dhcpc error</b>	Display error messages associated with the DHCP client.
<b>dhcpc packet</b>	Display packet information associated with the DHCP client.
<b>dhcpcd event</b>	Display event information associated with the DHCP server.
<b>dhcpcd packet</b>	Display packet information associated with the DHCP server.
<b>fover <i>option</i></b>	Display failover information. Refer to Table 5-6 for the <i>options</i> .
<b>h323</b>	Display information about the packet-based multimedia communications systems standard.
<b>h225 asn</b>	Display the output of the decoded PDUs.
<b>h225 events</b>	Display the events of the H225 signalling, or turn both traces on.
<b>h245 asn</b>	Display the output of the decoded PDUs.
<b>h245 events</b>	Display the events of the H245 signalling, or turn both traces on.
<b>ras asn</b>	Display the output of the decoded PDUs.
<b>ras events</b>	Display the events of the RAS signalling, or turn both traces on.
<b>icmp</b>	Display information about ICMP traffic.
<b>packet</b>	Display packet information.

<i>if_name</i>	Interface name from which the packets are arriving; for example, to monitor packets coming into the PIX Firewall from the outside, set <i>if_name</i> to <b>outside</b> .
<b>src</b> <i>source_ip</i>	Source IP address.
<b>netmask</b> <i>mask</i>	Network mask.
<b>dst</b> <i>dest_ip</i>	Destination IP address.
<b>proto icmp</b>	Display ICMP packets only.
<b>proto tcp</b>	Display TCP packets only.
<b>sport</b> <i>src_port</i>	Source port. See the “Ports” section in Chapter 1, “Introduction” for a list of valid port literal names.
<b>dport</b> <i>dest_port</i>	Destination port.
<b>proto udp</b>	Display UDP packets only.
<b>rx</b>	Display only packets received at the PIX Firewall.
<b>tx</b>	Display only packets that were transmitted from the PIX Firewall.
<b>both</b>	Display both received and transmitted packets.
<b>sqlnet</b>	Debug SQL*Net traffic.
<b>ppp</b>	Debug PPTP traffic, which is configured with the <b>vpdn</b> command.
<b>ppp error</b>	Display PPTP PPP virtual interface error messages.
<b>ppp io</b>	Display the packet information for the PPTP PPP virtual interface.
<b>ppp uauth</b>	Display the PPTP PPP virtual interface AAA user authentication debugging messages.
<b>ssh</b>	Debug information and error messages associated with the <b>ssh</b> command.
<b>vpdn event</b>	Display PPTP tunnel event change information.
<b>vpdn error</b>	Display PPTP protocol error messages.
<b>vpdn packet</b>	Display PPTP packet information about PPTP traffic.

#### Usage Guidelines

The **debug** command lets you view debug information. The **show debug** command displays the current state of tracing. You can debug the contents of network layer protocol packets with the **debug packet** command.

The **debug dhcpc detail** command displays detailed packet information about the DHCP client. The **debug dhcpc error** command displays DHCP client error messages. The **debug dhcpc packet** command displays packet information about the DHCP client. Use the **no** form of the **debug dhcpc** command to disable debugging.

The **debug dhcpcd event** command displays event information about the DHCP server. The **debug dhcpcd packet** command displays packet information about the DHCP server. Use the **no** form of the **debug dhcpcd** commands to disable debugging.

The **debug h323** command lets you debug H323 connections. Use the **no** form of the command to disable debugging. This command works when the **fixup protocol h323** command is enabled.


**Note**

The **debug h323** command, particularly the **debug h323 h225 asn**, **debug h323 h245 asn**, and **debug h323 ras asn** commands, might delay the sending of messages and cause slower performance in a real-time environment.

The **debug icmp trace** command shows ICMP packet information, the source IP address, and the destination address of packets arriving, departing, and traversing the PIX Firewall including pings to the PIX Firewall unit's own interfaces.

The **debug sqlnet** command reports on traffic between Oracle SQL\*Net clients and servers through the PIX Firewall.

The **debug ssh** command reports on information and error messages associated with the **ssh** command.

The **debug ppp** and **debug vpdn** commands provide information about PPTP traffic. PPTP is configured with the **vpdn** command.

Use of the **debug** commands can slow down busy networks.

For information about the **debug crypto** commands or IPSec-related debug commands, refer to the **debug** command page within the "Command Reference" chapter of the *IPSec User Guide for the Cisco Secure PIX Firewall Version 5.2*.

Table 5-6 lists the options for the **debug fover** command.

**Table 5-6** *debug fover Options*

Option	Description
<b>cable</b>	Failover cable status
<b>fail</b>	Failover internal exception
<b>fmsg</b>	Failover message
<b>get</b>	IP network packet received
<b>ifc</b>	Network interface status trace
<b>open</b>	Failover device open
<b>put</b>	IP network packet transmitted
<b>rx</b>	Failover cable receive
<b>rxdump</b>	Cable recv message dump (serial console only)
<b>rxip</b>	IP network failover packet received
<b>tx</b>	Failover cable transmit
<b>txdump</b>	Cable xmit message dump (serial console only)

Table 5-6 *debug fover Options (continued)*

Option	Description
<b>txip</b>	IP network failover packet transmit
<b>verify</b>	Failover message verify
<b>switch</b>	Failover Switching status

#### Trace Channel Feature

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

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

The location of the Trace Channel depends on whether you have a simultaneous Telnet console session running at the same time as the console session, or if you are using only the PIX Firewall serial console:

- If you are only using the PIX Firewall serial console, all **debug** commands display on the serial console.
- If you have both a serial console session and a Telnet console session accessing the console, then no matter where you enter the **debug** commands, the output displays on the Telnet console session.
- If you have two or more Telnet console sessions, the first session is the Trace Channel. If that session closes, the serial console session becomes the Trace Channel. The next Telnet console session that accesses the console will then become the Trace Channel.

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



#### Note

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

#### Additional debug Command Information



#### Note

Use of the **debug packet** command on a PIX Firewall experiencing a heavy load may result in the output displaying so fast that it may be impossible to stop the output by entering the **no debug packet** command from the console. You can enter the **no debug packet** command from a Telnet session.



#### Note

To let users ping through the PIX Firewall, add the **access-list *acl\_grp* permit icmp any any** command statement to the configuration and bind it to each interface you want to test with the **access-group** command. This lets pings go outbound and inbound.

To stop a **debug packet trace** command, enter:

```
no debug packet if_name
```

Replace *if\_name* with the name of the interface; for example, **inside**, **outside**, or a perimeter interface name.

To stop a **debug icmp trace** command, enter:

```
no debug icmp trace
```

### Examples

The following is partial sample output from the **debug dhcpc packet** and the **debug dhcpc detail** commands. The **ip address dhcp setroute** command was configured after turning on the **debug dhcpc** commands to obtain debugging information:

```
debug dhcpc packet
debug dhcpc detail
ip address outside dhcp setroute

DHCP:allocate request
DHCP:new entry. add to queue
DHCP:new ip lease str = 0x80ce8a28
DHCP:SDiscover attempt # 1 for entry:
Temp IP addr:0.0.0.0 for peer on Interface:outside
Temp sub net mask:0.0.0.0
    DHCP Lease server:0.0.0.0, state:1 Selecting
    DHCP transaction id:0x8931
    Lease:0 secs, Renewal:0 secs, Rebind:0 secs
    Next timer fires after:2 seconds
    Retry count:1 Client-ID:cisco-0000.0000.0000-outside

DHCP:SDiscover:sending 265 byte length DHCP packet
DHCP:SDiscover 265 bytes
DHCP Broadcast to 255.255.255.255 from 0.0.0.0
DHCP client msg received, fip=10.3.2.2, fport=67
DHCP:Received a BOOTREP pkt
DHCP:Scan:Message type:DHCP Offer
DHCP:Scan:Server ID Option:10.1.1.69 = 450A44AB
DHCP:Scan:Server ID Option:10.1.1.69 = 450A44AB
DHCP:Scan:Lease Time:259200
DHCP:Scan:Subnet Address Option:255.255.254.0
DHCP:Scan:DNS Name Server Option:10.1.1.70, 10.1.1.140
DHCP:Scan:Domain Name:example.com
DHCP:Scan:NBNS Name Server Option:10.1.2.228, 10.1.2.87
DHCP:Scan:Router Address Option:10.3.2.1
DHCP:rcvd pkt source:10.3.2.2, destination: 255.255.255.255
...
```

The following example turns on this command:

```
debug icmp trace
```

When you ping a host through the PIX Firewall from any interface, trace output displays on the console. The following example shows a successful ping from an external host (209.165.201.2) to the PIX Firewall unit's outside interface (209.165.201.1):

```
Inbound ICMP echo reply (len 32 id 1 seq 256) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 512) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 512) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 768) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 768) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 1024) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 1024) 209.165.201.1 > 209.165.201.2
NO DEBUG ICMP TRACE
ICMP trace off
```

This example shows that the ICMP packet length is 32 bytes, that the ICMP packet identifier is 1, and the ICMP sequence number. The ICMP sequence number starts at 0 and is incremented each time a request is sent.

The following is sample output from the **show debug** command output:

```
show debug
debug ppp error
debug vpdn event
debug crypto ipsec 1
debug crypto isakmp 1
debug crypto ca 1
debug icmp trace
debug packet outside both
debug sqlnet
```

The above sample output includes the **debug crypto** commands. Refer to the **debug** command page within the "Command Reference" chapter of the *IPSec User Guide for the Cisco Secure PIX Firewall Version 5.2* for more information about the **debug crypto** commands.

You can debug the contents of packets with the **debug packet** command:

```
debug packet inside
----- PACKET -----
-- IP --
4.3.2.1 ==> 255.3.2.1
   ver = 0x4      hlen = 0x5      tos = 0x0      tlen = 0x60
   id = 0x3902    flags = 0x0    frag off=0x0
   ttl = 0x20     proto=0x11    chksum = 0x5885
-- UDP --
      source port = 0x89      dest port = 0x89
      len = 0x4c      checksum = 0xa6a0
-- DATA --
00000014:                                00 01 00 00 |
....
00000024: 00 00 00 01 20 45 49 45 50 45 47 45 47 45 46 46 | ..
.. EIEPEGEGEFF
00000034: 43 43 4e 46 41 45 44 43 41 43 41 43 41 43 41 43 | CC
NFAEDCACACACAC
00000044: 41 43 41 41 41 00 00 20 00 01 c0 0c 00 20 00 01 | AC
AAA.. .....
00000054: 00 04 93 e0 00 06 60 00 01 02 03 04 00          | ..
.....
----- END OF PACKET -----
```

This display lists the information as it appears in a packet.

The following is sample output from the **show debug** command:

```
show debug
debug icmp trace off
debug packet off
debug sqlnet off
```

# dhcpd

This implements the DHCP server feature. (Configuration Mode)

**dhcpd address** *ip1[-ip2]* [*if\_name*]

**no dhcpd address** *ip1[-ip2]* [*if\_name*]

**dhcpd dns** *dns1* [*dns2*]

**no dhcpd dns** *dns1* [*dns2*]

**dhcpd wins** *wins1* [*wins2*]

**no dhcpd wins** *wins1* [*wins2*]

**dhcpd lease** *lease\_length*

**no dhcpd lease** *lease\_length*

**dhcpd domain** *domain\_name*

**no dhcpd domain** *domain\_name*

**dhcpd enable** [*if\_name*]

**no dhcpd enable** [*if\_name*]

**show dhcpd** [*binding|statistics*]

**clear dhcpd** [*binding|statistics*]

**debug dhcpd event**

**no debug dhcpd event**

**debug dhcpd packet**

**no debug dhcpd packet**

## Syntax Description

<b>address</b> <i>ip1</i> [ <i>ip2</i> ]	The IP pool address range. The size of the pool is limited to 10 addresses.
<i>if_name</i>	Name of the PIX Firewall interface. The default is the <b>inside</b> interface. Currently, the PIX Firewall DHCP server daemon can only be enabled on the <b>inside</b> interface.
<b>dns</b> <i>dns1</i> [ <i>dns2</i> ]	The IP addresses of the DNS servers for the DHCP client. The second server address is optional.
<b>wins</b> <i>wins1</i> [ <i>wins2</i> ]	The IP addresses of the Microsoft NetBios name servers (WINS server). The second server address is optional.

<b>lease</b> <i>lease_length</i>	The length of the lease in seconds granted to DHCP client from the DHCP server. The lease indicates how long the client can use the assigned IP address. The default is 3,600 seconds. The minimum lease length is 300 seconds, and the maximum lease length is 2,147,483,647 seconds.
<b>domain</b> <i>domain_name</i>	The DNS domain name. For example, <b>example.com</b> .
<b>binding</b>	The binding information for a given server IP address and its associated client hardware address and lease length.
<b>statistics</b>	Statistical information, such as address pool, number of bindings, malformed messages, sent messages, and received messages.

### Usage Guidelines

A DHCP Server provides network configuration parameters to a DHCP client. Support for the DHCP server within the PIX Firewall means the PIX Firewall can use the DHCP to configure connected PC clients. This DHCP feature is designed for the remote home or branch office that will establish a connection to a enterprise or corporate network. See “DHCP Server” within the Chapter 3, “Advanced Configurations” for information on how to implement the DHCP server feature into the PIX Firewall.



#### Note

The PIX Firewall DHCP server does not support **BOOTP** requests and **failover** configurations.

The **dhcpcd address** command specifies the DHCP server address pool. The address pool of a PIX Firewall DHCP server must be within the same subnet of the PIX Firewall interface that is enabled. In other words, the client must be physically connected to the subnet of a PIX Firewall interface. The size of the pool is currently limited to 10 addresses. The default for the PIX Firewall interface name is the **inside** interface, which is the only interface currently supported. The **no dhcpcd address** command removes the DHCP server address pool you configured.

The **dhcpcd dns** command specifies the IP address(es) of DNS server(s) for DHCP client. You have the option to specify two DNS servers. The **no dhcpcd dns** command removes the DNS IP address(es) from your configuration.

The **dhcpcd wins** command specifies the addresses of the WINS server for the DHCP client. The **no dhcpcd dns** command removes the WINS server IP address(es) from your configuration.

The **dhcpcd lease** command specifies the length of the lease in seconds granted to the DHCP client. This lease indicates how long the DHCP client can use the assigned IP address the DHCP granted. The **no dhcpcd lease** command removes the lease length that you specified from your configuration and replaces this value with the default value of 3,600 seconds.

The **dhcpcd domain** command specifies the DNS domain name for the DHCP client. For example, **example.com**. The **no dhcpcd domain** command removes the DNS domain server from your configuration.

The **dhcpcd enable** command enables the DHCP daemon to begin to listen for the DHCP client requests on the DHCP-enabled interface. The **no dhcpcd enable** command disables the DHCP server feature on the specified interface.

**Note**

In version 5.2, the PIX Firewall DHCP server daemon can only be enabled on the **inside** interface.

The **show dhcpd** command displays **dhcpd** commands, binding and statistics information associated with all of the **dhcpd** commands.

The **clear dhcpd** command clears all of the **dhcpd** commands, binding, and statistics information.

The **debug dhcpd event** command displays event information about the DHCP server. The **debug dhcpd packet** command displays packet information about the DHCP server. Use the **no** form of the **debug dhcpd** commands to disable debugging.

**Examples**

The following partial configuration example shows use of the **dhcpd address**, **dhcpd dns**, and **dhcpd enable** commands. In this example, an address pool for the DHCP clients is defined, a DNS server address is specified for the DHCP client, and the inside interface of the PIX Firewall is enabled for the DHCP server function:

```
dhcpd address 10.0.1.100-10.0.1.108
dhcpd dns 209.165.200.226
dhcpd enable
```

The following partial configuration example shows how to use three new version 5.2 PIX Firewall features that are associated with each other: DHCP server, DHCP client, and PAT using interface IP to configure a PIX Firewall in a small office, home office (SOHO) environment:

```
! use dhcp to configure the outside interface and default route
ip address outside dhcp setroute
! enable dhcp server daemon on the inside interface
ip address inside 10.0.1.2 255.255.255.0
dhcpd address 10.0.1.101-10.0.1.110
dhcpd dns 209.165.201.2 209.165.202.129
dhcpd wins 209.165.201.5
dhcpd lease 3000
dhcpd domain example.com
dhcpd enable
! use outside interface IP as PAT global address
nat (inside) 1 0 0
global (outside) 1 interface
```

The following is sample output for the **show dhcpd** command:

```
show dhcpd

dhcpd address 10.0.1.100-10.0.1.108 inside
dhcpd dns 192.23.21.23
dhcpd enable inside
```

The following is sample output for the **show dhcpd binding** command:

```
show dhcpd binding

IP Address Hardware Address Lease Expiration Type
10.0.1.100 0100.a0c9.868e.43 84985 seconds automatic
```

The following is sample output for the **show dhcpd statistics** command:

**show dhcpd statistics**

```
Address Pools 1
Automatic Bindings 1
Expired Bindings 1
Malformed messages 0
```

Message Received

```
BOOTREQUEST 0
DHCPDISCOVER 1
DHCPREQUEST 2
DHCPDECLINE 0
DHCPRELEASE 0
DHCPINFORM 0
```

Message Sent

```
BOOTREPLY 0
DHCPPOFFER 1
DHCPACK 1
DHCPNAK 1
```

# disable

Exit privileged mode and return to unprivileged mode. (Privileged mode.)

## **disable**

### Usage Guidelines

The **disable** command exits privileged mode and returns you to unprivileged mode. Use the **enable** command to return to privileged mode.

### Examples

The following example shows how to exit privileged mode:

```
pixfirewall# disable  
pixfirewall>
```

# enable

Start privileged mode. (Unprivileged mode.)

## enable

### Usage Guidelines

The **enable** command starts privileged mode. The PIX Firewall prompts you for your privileged mode password. By default, a password is not required—press the **Enter** key at the Password prompt to start privileged mode. Use **disable** to exit privileged mode. Use **enable password** to change the password.

### Examples

The following example shows how to start privileged mode with the **enable** command and then configuration mode with the **configure terminal** command.

```
pixfirewall> enable
Password:
pixfirewall# configure terminal
pixfirewall(config)#
```

# enable password

Set the privileged mode password. (Privileged mode.)

**enable password** *password* [**encrypted**]

**show enable password**

## Syntax Description

*password* A case-sensitive password of up to 16 alphanumeric characters.

**encrypted** Specifies that the password you entered is already encrypted. The *password* must be 16 characters in length.

## Usage Guidelines

The **enable password** command changes the privileged mode password, for which you are prompted after you enter the **enable** command. When the PIX Firewall starts and you enter privileged mode, the password prompt appears. There is not a default password (press the **Enter** key at the Password prompt). The **show enable password** command lists the encrypted form of the password.

You can return the **enable password** to its original value (press the **Enter** key at prompt) by entering:

```
pixfirewall# enable password
pixfirewall#
```



### Note

---

If you change the password, write it down and store it in a manner consistent with your site's security policy. Once you change this password, you cannot view it again. Also, ensure that all who access the PIX Firewall console are given this password.

---

Use the **passwd** command to set the password for PIX Firewall Manager and Telnet access to the PIX Firewall console. The default **passwd** value is **cisco**.

See also: **passwd**.

### Examples

The following examples show how to start privileged mode with the **enable** command, change the enable password with the **enable password** command, enter configuration mode with the **configure terminal** command, and display the contents of the current configuration with the **write terminal** command:

```
pixfirewall> enable
Password:
pixfirewall# enable password w0ttal1fe
pixfirewall# configure terminal
pixfirewall(config)# write terminal
Building configuration...
...
enable password 2oifudsaoiD.9ff encrypted
...
```

The following example shows the use of the **encrypted** option:

```
enable password 1234567890123456 encrypted
show enable password
enable password 1234567890123456 encrypted

enable password 1234567890123456
show enable password
enable password feCkwUGktTCAGIbD encrypted
```

# established

Permit return connections on ports other than those used for the originating connection based on an established connection. (Configuration mode.)

**established** *protocol src\_port [dest\_port] [permitto protocol dport[-dport]] [permitfrom protocol sport[-sport]]*

**no established** *protocol src\_port [dest\_port] [permitto protocol dport[-dport]] [permitfrom protocol sport[-sport]]*

**clear established**

**show established**

## Syntax Description

<i>src_port</i>	The source port used for the established connection lookup. This is the originating traffic's source port and may be specified as 0 if the protocol does not specify which source port(s) will be used. Use wildcard ports (0) only when necessary.
<i>dest_port</i>	The destination port used for the established connection lookup. This is the originating traffic's destination port and may be specified as 0 if the protocol does not specify which destination port(s) will be used. Use wildcard ports (0) only when necessary.
<b>permitto</b>	Used to specify the return traffic's protocol and to which destination port(s) the traffic will be permitted.
<b>permitfrom</b>	Used to specify the return traffic's protocol and from which source port(s) the traffic will be permitted.
<i>sport</i>	The source port(s) from which the return traffic is permitted.
<i>dport</i>	The destination port(s) to which the return traffic is permitted.

## Usage Guidelines

The **established** command allows outbound connections return access through the PIX Firewall. This command works with two connections, an original connection outbound from a network protected by the PIX Firewall and a return connection inbound between the same two devices on an external host.

The first protocol, destination port and optional source port specified is for the initial outbound connection. The **permitto** and **permitfrom** options refine the return inbound connection.



### Note

Cisco recommends that you always specify the **established** command with the **permitto** and **permitfrom** options. Without these options, the use of the **established** command opens a security hole that can be exploited for attack of your internal systems. See the “Security Problem” section that follows for more information.

The **permitto** option lets you specify a new protocol or port for the return connection at the PIX Firewall.

The **permitfrom** option lets you specify a new protocol or port at the remote server.

The **no established** command disables the **established** feature.

The **show established** command shows the **established** commands in the configuration.

The **clear established** command removes all **establish** command statements from your configuration.



Note

For the **established** command to work properly, the client must listen on the port specified with the **permitto** option.

You can use the established command with the nat 0 command statement (where there are no global command statements).



Note

The established command cannot be used with PAT (Port Address Translation).

The **established** command works as shown in the following format:

```
established A B permitto C D permitfrom E F
```

This command works as though it were written “If there exists a connection between two hosts using protocol A on ports B and C, permitting return connections through the PIX Firewall via protocol D, if the destination port(s) correspond to E (protocols D and F must match, but can be different than A), and the source port(s) correspond to G.”

For example:

```
established tcp 6060 0 permitto tcp 6061 permitfrom tcp 6059
```

In this case, if a connection is started by an internal host to an external host using TCP source port 6060 and any destination port, the PIX Firewall permits return traffic between the hosts via TCP destination port 6061 and TCP source port 6059.

For example:

```
established udp 0 6060 permitto tcp 6061 permitfrom tcp 1024-65535
```

In this case, if a connection is started by an internal host to an external host using UDP destination port 6060 and any source port, the PIX Firewall permits return traffic between the hosts via TCP destination port 6061 and TCP source port 1024-65535.

### Security Problem

The **established** command has been enhanced to optionally specify the destination port used for connection lookups. Only the source port could be specified previously with the destination port being 0 (a wildcard). This addition allows more control over the command and provides support for protocols where the destination port is known, but the source port is not.

The **established** command can potentially open a large security hole in the PIX Firewall if not used with discretion. Whenever you use this command, if possible, also use the **permitto** and **permitfrom** options to indicate ports to which and from which access is permitted. Without these options, external systems to which connections are made could make unrestricted connections to the internal host involved in the connection. The following are examples of potentially serious security violations that could be allowed when using the **established** command.

Example:

```
established tcp 0 4000
```

With this example, if an internal system makes a TCP connection to an external host on port 4000, then the external host could come back in on any port using any protocol.

Example:

```
established tcp 0 0 (same as previous releases established tcp 0)
```

With this example, if something like the following exists:

```
static (inside,outside) 200.0.0.2 10.0.0.2
access-list acl_grp permit tcp host 200.0.0.2 eq www any
```

an attacker only need make a web connection to 200.0.0.2 and then they can make unrestricted connections using any protocol or ports.

### Examples

The following example occurs when a local host 10.1.1.1 starts a TCP connection on port 9999 to a foreign host 209.165.201.1. The example allows packets from the foreign host 209.165.201.1 on port 4242 back to local host 10.1.1.1 on port 5454:

```
established tcp 9999 permitto tcp 5454 permitfrom tcp 4242
```

The next example allows packets from foreign host 209.165.201.1 on any port back to local host 10.1.1.1 on port 5454:

```
established tcp 9999 permitto tcp 5454
```

### XDMCP Support

PIX Firewall now provides support for XDMCP (X Display Manager Control Protocol) with assistance from the **established** command.



Note

---

XDMCP is on by default, but will not complete the session unless the **established** command is used.

---

Example:

```
established tcp 0 6000 to tcp 6000 from tcp 1024-65535
```

Will allow internal XDMCP equipped (UNIX or ReflectionX) hosts to access external XDMCP equipped XWindows servers. UDP/177 based XDMCP negotiates a TCP based XWindows session and subsequent TCP back connections will be permitted. Because the source port(s) of the return traffic is unknown, the *src\_port* field should be specified as 0 (wildcard). The destination port, *dest\_port*, will typically be 6000; the well-known XServer port. The *dest\_port* should be 6000 + *n*; where *n* represents the local display number. Use the following UNIX command to change this value:

```
setenv DISPLAY hostname:displaynumber.screennumber
```

The **established** command is needed because many TCP connections are generated (based on user interaction) and the source port for these connection is unknown. Only the destination port will be static. The PIX Firewall does XDMCP fixups transparently. No configuration is required, but the **established** command is necessary to accommodate the TCP session. Be advised that using applications like this through the PIX Firewall may open up security holes. The XWindows system has been exploited in the past and newly introduced exploits are likely to be discovered.

# exit

Exit an access mode. (All modes.)

**exit**

## Usage Guidelines

Use the **exit** command to exit from an access mode. This command is the same as **quit**.

## Examples

The following example shows how to exit configuration mode and then privileged mode:

```
pixfirewall(config)# exit  
pixfirewall# exit  
pixfirewall>
```

# failover

Change or view access to the optional failover feature. (Configuration mode.)

**failover** [**active**]

**failover ip address** *if\_name ip\_address*

**failover link** [*stateful\_if\_name*]

**failover poll** *seconds*

**failover reset**

**no failover active**

**show failover**

## Syntax Description

<b>active</b>	Make a PIX Firewall the Active unit. Use this command when you need to force control of the connection back to the unit you are accessing, such as when you want to switch control back from a unit after you have fixed a problem and want to restore service to the Primary unit. Either enter <b>no failover active</b> on the secondary unit to switch service to the primary or <b>failover active</b> on the Primary unit.
<i>if_name</i>	Interface on which the Standby unit resides.
<i>ip_address</i>	The IP address used by the Standby unit to communicate with the Active unit. Use this IP address with the <b>ping</b> command to check the status of the Standby unit. This address must be on the same network as the system IP address. For example, if the system IP address is 192.159.1.3, set the failover IP address to 192.159.1.4.
<b>link</b>	Specify the interface where a fast LAN link is available for Stateful Failover.
<i>stateful_if_name</i>	In addition to the failover cable, a dedicated fast LAN link is required to support Stateful Failover. Do not use FDDI because of its blocksize or Token Ring because Token Ring requires additional time to insert into the ring. The default interface is the highest LAN port with failover configured.

- poll *seconds*** Specify how long failover waits before sending special failover “hello” packets between the Primary and Standby units over all network interfaces and the failover cable. The default is 15 seconds. The minimum value is 3 seconds and the maximum is 15 seconds. Set to a lower value for Stateful Failover. With a faster poll time, PIX Firewall can detect failure and trigger failover faster. However, faster detection may cause unnecessary switchovers when the network is temporarily congested or a network card starts slowly.
- reset** Force both units back to an unfailed state. Use this command once the fault has been corrected. The **failover reset** command can be entered from either unit, but it is best to always enter commands at the Active unit. Entering the **failover reset** command at the Active unit will “unfail” the Standby unit.

### Usage Guidelines

Use the **failover** command without an argument after you connect the optional failover cable between your primary firewall and a secondary firewall. The default configuration has failover enabled. Enter **no failover** in the configuration file for the PIX Firewall if you will not be using the failover feature. Use the **show failover** command to verify the status of the connection and to determine which unit is active.



Note

See “Failover” in Chapter 3, “Advanced Configurations,” for configuration information.



Note

For Failover, PIX Firewall requires any unused interfaces be given IP addresses and connected to the Standby unit for use in receiving Failover checkup messages.



Note

Set the Stateful Failover dedicated interface to 100 Mbps full duplex using the **100full** option to the **interface** command.

Use the **failover active** command to initiate a failover switch from the Standby unit, or the **no failover active** command from the Active unit to initiate a failover switch. You can use this feature to return a failed unit to service, or to force an Active unit offline for maintenance. Because the Standby unit does not keep state information on each connection, all active connections will be dropped and must be re-established by the clients.

Use the **failover link** command to enable Stateful Failover. The Stateful Failover interface can be either Ethernet or Token Ring interfaces. FDDI interfaces are supported for non-Stateful Failover interfaces.

If a failover IP address has not been entered, **show failover** will display 0.0.0.0 for the IP address, and monitoring of the interfaces will remain in “waiting” state. A failover IP address must be set for failover to work.

The **failover poll *seconds*** command lets you determine how long failover waits before sending special failover “hello” packets between the Primary and Standby units over all network interfaces and the failover cable. The default is 15 seconds. The minimum value is 3 seconds and the maximum is 15 seconds. Set to a lower value for Stateful Failover. With a faster poll time, PIX Firewall can detect failure and trigger failover faster. However, faster detection may cause unnecessary switchovers when the network is temporarily congested or a network card starts slowly.

When a failover cable connects two PIX Firewall units, the **no failover** command now disables failover until you enter the **failover** command to explicitly enable failover. Previously, when the failover cable connected two PIX Firewall units and you entered the **no failover** command, failover would automatically re-enable after 15 seconds.

If you reboot the PIX Firewall without entering the **write memory** command and the failover cable is connected, failover mode automatically enables.

You can also view the information from the **show failover** command using SNMP. Refer to “Using the Firewall and Memory Pool MIBs” in Chapter 3, “Advanced Configurations,” for more information.

A failover configuration example is provided in “Failover Configuration” in Chapter 4, “Configuration Examples.”

### Examples

The following output shows that failover is enabled, and that the Primary unit state is active:

```
show failover
Failover On
Cable status: Normal
Reconnect time-out 0:00:00
  This host: Primary - Active
    Active time: 3456 (sec)
    Interface 4th (172.16.1.112): Normal
    Interface intf3 (192.168.3.2): Normal
    Interface intf2 (192.168.2.2): Normal
    Interface outside (192.168.1.8): Normal
    Interface inside (10.1.1.6): Normal
  Other host: Secondary - Standby
    Active time: 0 (sec)
    Interface 4th (172.16.1.111): Normal
    Interface intf3 (192.168.3.1): Normal
    Interface intf2 (192.168.2.1): Normal
    Interface outside (192.168.1.7): Normal
    Interface inside (10.1.1.2): Normal

Standby Logical Update Statistics
Link : intf2
Stateful Obj   xmit   xerr   rcv    rerr
General        53     0      0      0
sys cmd        53     0      0      0
up time         0      0      0      0
xlate           0      0      0      0
tcp conn       0      0      0      0
udp conn       0      0      0      0
ARP tbl        0      0      0      0
RIF Tbl        0      0      0      0
```

The “Cable status” has these values:

- Normal—Indicates that the Active unit is working and that the Standby unit is ready.
- Waiting—Indicates that monitoring of the other unit’s network interfaces has not yet started.
- Failed—Indicates that the PIX Firewall has failed.

You can view the IP addresses of the Standby unit with the **show ip address** command:

**show ip address**

System IP Addresses:

```
ip address outside 209.165.201.2 255.255.255.224
ip address inside 192.168.2.1 255.255.255.0
ip address perimeter 192.168.70.3 255.255.255.0
```

Current IP Addresses:

```
ip address outside 209.165.201.2 255.255.255.224
ip address inside 192.168.2.1 255.255.255.0
ip address perimeter 192.168.70.3 255.255.255.0
```

The Current IP Addresses are the same as the System IP Addresses on the failover Active unit. When the Primary unit fails, the Current IP Addresses become those of the Standby unit.

The Standby Logical Update Statistics output that displays when you use the **show failover** command only describes Stateful Failover. The “xerrs” value does not indicate an error in failover and can be ignored.

# filter

Enable or disable outbound URL or HTML object filtering. (Configuration mode.)

**filter activex** *port local\_ip mask foreign\_ip mask*

**no filter activex** *port local\_ip mask foreign\_ip mask*

**filter java** *port[-port] local\_ip mask foreign\_ip mask*

**no filter java** *port[-port] local\_ip mask foreign\_ip mask*

**filter url** *port|except local\_ip local\_mask foreign\_ip foreign\_mask [allow]*

**no filter url** *port | except [local\_ip local\_mask foreign\_ip foreign\_mask]*

**clear filter**

**show filter**

## Syntax Description

<b>activex</b>	Block outbound ActiveX, Java applets, and other HTML <object> tags from outbound packets.
<b>java</b>	Block Java applets returning to the PIX Firewall as a result of an outbound connection.
<b>url</b>	Filter URLs (Universal Resource Locators) from data moving through the PIX Firewall.
<b>except</b>	<b>filter url</b> only: Create an exception to a previous <b>filter</b> condition.
<i>port</i>	The Web traffic port. Typically, this is port 80, but other values are accepted. The <b>http</b> literal can be used for port 80.
<i>port[-port]</i>	<b>filter java</b> only: One or more ports on which Java applets may be received.
<i>local_ip</i>	The IP address of the highest security level interface from which access is sought. You can set this address to <b>0.0.0.0</b> (or in shortened form, <b>0</b> ) to specify all hosts.
<i>local_mask</i>	Network mask of <i>local_ip</i> . You can use <b>0.0.0.0</b> (or in shortened form, <b>0</b> ) to specify all hosts.
<i>foreign_ip</i>	The IP address of the lowest security level interface to which access is sought. You can use <b>0.0.0.0</b> (or in shortened form, <b>0</b> ) to specify all hosts.

- foreign\_mask* Network mask of *foreign\_ip*. Always specify a specific mask value. You can use **0.0.0.0** (or in shortened form, **0**) to specify all hosts.
- allow** **filter url** only: When the server is unavailable, let outbound connections pass through PIX Firewall without filtering. If you omit this option, and if the Websense server goes offline, PIX Firewall stops outbound port 80 (Web) traffic until the Websense server is back online.

### Usage Guidelines

The sections that follow describe each type of filter. The **clear filter** command removes all **filter** commands from the configuration. The **show filter** command lists all **filter** commands in the configuration.

### filter activex

The **filter activex** command filters out ActiveX, Java applets, and other HTML <object> usages from outbound packets. ActiveX controls, formerly known as OLE or OCX controls, are components you can insert in a web page or other application. These controls include custom forms, calendars, or any of the extensive third-party forms for gathering or displaying information.

As a technology, it creates many potential problems for the network clients including causing workstations to fail, introducing network security problems, or be used to attack servers.

This feature blocks the HTML <object> tag and comments it out within the HTML web page.



#### Note

The <object> tag is also used for Java applets, image files, and multimedia objects, which will also be blocked by the **filter activex** command. If the <object> or </object> HTML tags split across network packets or if the code in the tags is longer than the number of bytes in the MTU, PIX Firewall cannot block the tag.



#### Note

ActiveX blocking does not occur when users access an IP address referenced by the **alias** command.

### Examples

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

```
filter activex 80 0 0 0 0
```

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

### filter java

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



#### Note

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

### Examples

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

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

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

### filter url

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

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



#### Note

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

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

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

Websense protocol version 4 contains the following enhancements:

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

Follow these steps to filter URLs:

- 
- Step 1** Designate a Websense server with the **url-server** command.
  - Step 2** Enable filtering with the **filter** command.
  - Step 3** If needed, improve throughput with the **url-cache** command. However, this command does not update Websense logs, which may affect Websense accounting reports. Accumulate Websense run logs before using the **url-cache** command.
  - Step 4** Use the **show url-cache stats** and the **show perfmon** commands to view run information.
- 

Information on Websense is available at the following site:

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

### Examples

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

```
url-server (perimeter) host 10.0.1.1
filter url 80 0 0 0 0
filter url except 10.0.2.54 255.255.255.255 0 0
```

The following example filters all outbound HTTP connections received from a proxy server that sends Web traffic on port 8080:

```
filter url 8080 0 0 0 0
```

# fixup protocol

Change, enable, disable, or list a PIX Firewall application protocol feature. (Configuration mode.)

```

fixup protocol ftp [strict] [port]
fixup protocol http [port[-port]]
fixup protocol h323 [port[-port]]
fixup protocol rsh [514]
fixup protocol rtsp [port]
fixup protocol sip [5060]
fixup protocol smtp [port[-port]]
fixup protocol sqlnet [port[-port]]
no fixup protocol protocol [port[-port]]
clear fixup
show fixup [protocol protocol]

```

## Syntax Description

<b>protocol</b>	Specify the protocol to fix up: <b>ftp</b> , <b>http</b> , <b>h323</b> , <b>rsh</b> , <b>rtsp</b> , <b>sip</b> , <b>smtp</b> , or <b>sqlnet</b> .
<i>port</i>	Specify the port number or range for the application protocol. The default ports are: TCP 21 for <b>ftp</b> , TCP 80 for <b>http</b> , TCP 1720 for <b>h323</b> , TCP 514 for <b>rsh</b> , TCP 554 for <b>rtsp</b> , TCP 25 for <b>smtp</b> , TCP 1521 for <b>sqlnet</b> , and TCP 5060 for <b>sip</b> . The default port value for <b>rsh</b> cannot be changed, but additional port statements can be added. See the “Ports” section in Chapter 1, “Introduction” for a list of valid port literal names.
<b>strict</b>	Prevent web browsers from sending embedded commands in FTP requests. Each FTP command must be acknowledged before a new command is allowed. Connections sending embedded commands are dropped.

## Usage Guidelines

The **fixup protocol** commands let you view, change, enable, or disable the use of a service or protocol through the PIX Firewall. The ports you specify are those that the PIX Firewall listens at for each respective service. You can change the port value for each service except **rsh** and **sip**. The **fixup protocol** commands are always present in the configuration and are enabled by default.

The **fixup protocol** command performs the Adaptive Security Algorithm based on different port numbers other than the defaults. This command is global and changes things for both inbound and outbound connections, and cannot be restricted to any **static** command statements.

The **clear fixup** command removes **fixup** commands from the configuration that you added. It does not remove the default **fixup protocol** commands.

The **show fixup** command lists all values or the **show fixup protocol *protocol*** command lists an individual protocol.

You can disable the fixup of a protocol by removing all fixups of the protocol from the configuration using the **no fixup** command. After you remove all fixups for a protocol, the **no fixup** form of the command or the default port is stored in the configuration.

The following lists the default **fixup protocol** values (those enabled when a PIX Firewall is first installed). You can view the **fixup protocol** settings with the **show fixup** command as follows:

```
show fixup
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 1720
fixup protocol rsh 514
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
```

### fixup protocol ftp

The FTP port can be changed; however if you change the default of port 21, to something like 2021, all FTP control connections must happen on port 2021. FTP control connections on port 21 will no longer work.

If you disable FTP fixups with the **no fixup protocol ftp** command, outbound users can start connections only in passive mode, and all inbound FTP is disabled.

The **strict** option to the **fixup protocol ftp** command prevents web browsers from sending embedded commands in FTP requests. Each FTP command must be acknowledged before a new command is allowed. Connections sending embedded commands are dropped. The **strict** option only lets an FTP server generate the 227 command and only lets an FTP client generate the PORT command. The 227 and PORT commands are checked to ensure they do not appear in an error string.

The *port* parameter lets you specify the port at which the PIX Firewall listens for FTP traffic. Typically, this value is 21. In addition, the FTP port can now only be in the range of 1 to 1024.

### fixup protocol h323

The **fixup protocol h323** command provides support for Intel InternetPhone, CU-SeeMe, CU-SeeMe Pro, MeetingPoint, and MS NetMeeting. Version 5.2 supports H.323 version 2. H.323 is a suite of protocols defined by the International Telecommunication Union (ITU) for multimedia conferences over LANs. H.323 supports VoIP gateways and VoIP gatekeepers. H.323 version 2 adds the following functionality to the PIX Firewall:

- Fast Connect or Fast Start Procedure for faster call setup
- H.245 tunneling for resource conservation, call synchronization, and reduced set up time

### fixup protocol http



#### Note

If there is a **no fixup protocol http** command statement in the configuration, the **filter url** command does not work.

### fixup protocol rtsp

The **fixup protocol rtsp** command lets PIX Firewall pass RTSP (Real Time Streaming Protocol) packets. RTSP is used by RealAudio, RealNetworks, Apple QuickTime 4, RealPlayer, and Cisco IP/TV connections. PIX Firewall does not support multicast RTSP.

If you are using Cisco IP/TV, use RTSP TCP port 554 and TCP 8554:

```
fixup protocol rtsp 554
fixup protocol rtsp 8554
```

The following restrictions apply to the **fixup protocol rtsp** command:

1. This PIX Firewall will not fix RTSP messages passing through UDP ports.
2. PIX Firewall does not support the RealNetwork's multicast mode (x-real-rdt/mcast).
3. PAT is not supported with the **fixup protocol rtsp** command.
4. PIX Firewall does not have the ability to recognize HTTP cloaking where RTSP messages are hidden in the HTTP messages.
5. PIX Firewall cannot perform NAT on RTSP messages because the embedded IP addresses are contained in the SDP files as part of HTTP or RTSP messages. Packets could be fragmented and PIX Firewall cannot perform NAT on fragmented packets.
6. With Cisco IP/TV, the number of NATs the PIX Firewall performs on the SDP part of the message is proportional to the number of program listings in the Content Manager (each program listing can have at least six embedded IP addresses).
7. You can configure NAT for Apple QuickTime 4 or RealPlayer; however, Cisco IP/TV will not work if both the Content Manager and the Server are inside relative to the PIX Firewall.
8. When using RealPlayer, it is important to properly configure transport mode. For the PIX Firewall, add an **access-list** command statement from the server to the client or vice versa. For RealPlayer, change transport mode by clicking **Options>Preferences>Transport>RTSP Settings**.

If using TCP mode on the RealPlayer, select the check boxes for **Use TCP to Connect to Server** and **Attempt to use TCP for all content**. On the PIX Firewall, there is no need to configure the **fixup**.

If using UDP mode on the RealPlayer, select the check boxes for **Use TCP to Connect to Server** and **Attempt to use UDP for static content**, and for live content not available via Multicast. On the PIX Firewall, add a **fixup protocol rtsp port** command statement

### fixup protocol sip

The **fixup protocol sip** command enables SIP on the interface. SIP enables call handling sessions—particularly two-party audio conferences, or “calls.” SIP works with Session Description Protocol (SDP) for call signalling. SDP specifies the ports for the media stream. Using SIP, the PIX Firewall can support any SIP Voice over IP (VoIP) gateways and VoIP proxy servers.

Session initiation protocol (SIP), as defined by the Internet Engineering Task Force (IETF), enables call handling sessions, particularly two-party audio conferences, or “calls.” SIP works with Session Description Protocol (SDP) for call signalling. SDP specifies the ports for the media stream. Using SIP, the PIX Firewall can support any SIP Voice over IP (VoIP) gateways and VoIP proxy servers. SIP and SDP are defined in the following RFCs:

- SIP: session initiation protocol, RFC 2543
- SDP: Session Description Protocol, RFC 232

### fixup protocol smtp

The **fixup protocol smtp** command enables the Mail Guard feature, which only lets mail servers receive the RFC 821, section 4.5.1 commands of HELO, MAIL, RCPT, DATA, RSET, NOOP, and QUIT. All other commands are rejected with the “500 command unrecognized” reply code.

## fixup protocol sqlnet



### Note

---

PIX Firewall uses port 1521 for SQL\*Net. This is the default port used by Oracle for SQL\*Net; however, this value does not agree with IANA port assignments.

---

### Examples

You can add multiple port settings for each protocol with separate commands; for example:

```
fixup protocol ftp 21
fixup protocol ftp 4254
fixup protocol ftp 9090
```

These commands cause PIX Firewall to listen to the standard FTP port of 21 but also to listen for FTP traffic at ports 4254 and 9090.

The following example enables access to an inside server running Mail Guard:

```
static (inside,outside) 209.165.201.1 192.168.42.1 netmask 255.255.255.255
access-list acl_out permit tcp host 209.165.201.1 eq smtp any
access-group acl_out in interface outside
fixup protocol smtp 25
```

The following example shows the commands to disable Mail Guard:

```
static (dmz1,outside) 209.165.201.1 10.1.1.1 netmask 255.255.255.255
access-list acl_out permit tcp host 209.165.201.1 eq smtp any
access-group acl_out in interface outside
no fixup protocol smtp 25
```

In this example, the **static** command sets up a global address to permit outside hosts access to the 10.1.1.1 mail server host on the dmz1 interface. (The MX record for DNS needs to point to the 209.165.201.1 address so that mail is sent to this address.) The **access-list** command lets any outside users access the global address through the SMTP port (25). The **no fixup protocol** command disables the Mail Guard feature.

# flashfs

Clear, display, or downgrade filesystem information. (Configuration mode.)

**flashfs downgrade** { **4.x** | **5.0** | **5.1** }

**clear flashfs**

**show flashfs**

## Syntax Description

- |                      |  |
|----------------------|--|
| <b>downgrade 4.x</b> | Clear the filesystem information from Flash memory before downgrading to version 4.0, 4.1, 4.2, 4.3, or 4.4. |
| <b>downgrade 5.0</b> | Write the filesystem to Flash memory before downgrading to version 5.0.                                      |
| <b>downgrade 5.1</b> | Write the filesystem to Flash memory before downgrading to version 5.1.                                      |

## Usage Guidelines

The **clear flashfs** and the **flashfs downgrade 4.x** commands clear the filesystem part of Flash memory in the PIX Firewall. Versions 4.*n* cannot use the information in the filesystem so it needs to be cleared to let the earlier version operate correctly.

The **flashfs downgrade 5.0 | 5.1** command reorganizes the filesystem part of Flash memory so that information stored in the filesystem can be accessed by the earlier version. The PIX Firewall maintains a filesystem in Flash memory to store system information, IPSec private keys, certificates, and CRLs. It is crucial that you clear or reformat the filesystem before downgrading to a previous PIX Firewall version. Otherwise, your filesystem will get out of sync with the actual contents of the Flash memory and cause problems when the unit is later upgraded.

You only need to use the **flashfs downgrade 5.0 | 5.1** command if your PIX Firewall has 16 MB Flash memory, if you have IPSec private keys, certificates, or CRLs stored in Flash memory, and you used the **ca save all** command to save these items in Flash memory. The **flashfs downgrade 5.0 | 5.1** command fails if the filesystem indicates that any part of the image, configuration, or private data in the Flash memory device is unusable.

The **clear flashfs** and **flashfs downgrade** commands do not affect the configuration stored in Flash memory.

The **clear flashfs** command is the same as the **flashfs downgrade 4.x** command.

The **show flashfs** command displays the size in bytes of each filesystem sector and the current state of the filesystem. The data in each sector is as follows:

- file 0—PIX Firewall binary image, where the .bin file is stored.
- file 1—PIX Firewall configuration data that you can view with the **show config** command.
- file 2—PIX Firewall datafile that stores IPSec key and certificate information.
- file 3—**flashfs downgrade** information for the **show flashfs** command.

### Examples

Use the following command to write the filesystem to Flash memory before downgrading to version 5.1:

```
flashfs downgrade 5.1
```

The following commands display the filesystem sector sizes:

```
show flashfs  
flash file system: version:1 magic:0x12345679  
  file 0: origin:      0 length:1794104  
  file 1: origin: 2095104 length:1496  
  file 2: origin:      0 length:0  
  file 3: origin: 2096640 length:140
```

```
flashfs downgrade 5.1
```

```
show flashfs  
flash file system: version:0 magic:0x0  
  file 0: origin:      0 length:0  
  file 1: origin:      0 length:0  
  file 2: origin:      0 length:0  
  file 3: origin:      0 length:0
```

The origin values are integer multiples of the underlying filesystem sector size.

# floodguard

Enable or disable Flood Defender to protect against flood attacks. (Configuration mode.)

**floodguard enable | disable**

**show floodguard**

## Syntax Description

**enable** Enable Flood Defender.

**disable** Disable Flood Defender.

## Usage Guidelines

The **floodguard** command lets you reclaim PIX Firewall resources if the user authentication (uauth) subsystem runs out of resources. If an inbound or outbound uauth connection is being attacked or overused, the PIX Firewall will actively reclaim TCP user resources.

When the resources deplete, the PIX Firewall lists messages about it being out of resources or out of tcpusers.

If the PIX Firewall uauth subsystem is depleted, TCP user resources in different states are reclaimed depending on urgency in the following order:

1. Timewait
2. FinWait
3. Embryonic
4. Idle

The **floodguard** command is enabled by default.

## Examples

The following example enables the **floodguard** command and lists the **floodguard** command statement in the configuration:

```
floodguard enable
show floodguard
floodguard enable
```

# global

Create or delete entries from a pool of global addresses. (Configuration mode.)

```
global [(if_name)] nat_id {global_ip [-global_ip] [netmask global_mask]} | interface
no global [(if_name)] nat_id [global_ip [-global_ip] [netmask global_mask]] | [interface]
show global
```

## Syntax Description

<i>if_name</i>	The external network where you use these global addresses.
<i>nat_id</i>	A positive number shared with the <b>nat</b> command that groups the <b>nat</b> and <b>global</b> command statements together. The valid ID numbers can be any positive number up to 2,147,483,647.
<i>global_ip</i>	One or more global IP addresses that the PIX Firewall shares among its connections. If the external network is connected to the Internet, each global IP address must be registered with the Network Information Center (NIC). You can specify a range of IP addresses by separating the addresses with a dash (-).  You can create a Port Address Translation (PAT) <b>global</b> command statement by specifying a single IP address. You can have one PAT <b>global</b> command statement per interface. A PAT can support up to 65,535 xlate objects.
<b>netmask</b>	Reserved word that prefaces the network <i>global_mask</i> variable.
<i>global_mask</i>	The network mask for <i>global_ip</i> . If subnetting is in effect, use the subnet mask; for example, 255.255.255.128. If you specify an address range that overlaps subnets, <b>global</b> will not use the broadcast or network addresses in the pool of global addresses. For example, if you use 255.255.255.224 and an address range of 209.165.201.1-209.165.201.30, the 209.165.201.31 broadcast address and the 209.165.201.0 network address will not be included in the pool of global addresses.
<b>interface</b>	Specifies PAT using the IP address at the interface.

## Usage Guidelines

The **global** command defines a pool of global addresses. The global addresses in the pool provide an IP address for each outbound connection, and for those inbound connections resulting from outbound connections. Ensure that associated **nat** and **global** command statements have the same *nat\_id*.

After changing or removing a **global** command statement, use the **clear xlate** command.

Use the **no global** command to remove access to a *nat\_id*, or to a Port Address Translation (PAT) address, or address range within a *nat\_id*. Use the **show global** command to view the **global** command statements in the configuration.

## Usage Notes

1. You can enable the PAT (Port Address Translation) feature by entering a single IP address with the **global** command. PAT lets multiple outbound sessions appear to originate from a single IP address. With PAT enabled, the firewall chooses a unique port number from the PAT IP address for each outbound xlate (translation slot). This feature is valuable when an Internet service provider cannot allocate enough unique IP addresses for your outbound connections. An IP address you specify for a PAT cannot be used in another global address pool.
2. When a PAT augments a pool of global addresses, first the addresses from the global pool are used, then the next connection is taken from the PAT address. If a global pool address frees, the next connection takes that address. The global pool addresses always come first, before a PAT address is used. Augment a pool of global addresses with a PAT by using the same *nat\_id* in the **global** command statements that create the global pools and the PAT. For example:

```
global (outside) 1 209.165.201.1-209.165.201.10 netmask 255.255.255.224
global (outside) 1 209.165.201.22 netmask 255.255.255.224
```

3. PAT does not work with H.323 applications and caching nameservers. Do not use a PAT when multimedia applications need to be run through the firewall. Multimedia applications can conflict with port mappings provided by PAT.
4. PAT does not work with the **established** command.
5. PAT works with DNS, FTP and passive FTP, HTTP, email, RPC, rshell, Telnet, URL filtering, and outbound traceroute.

However for use with passive FTP, use the **fixup protocol ftp strict ftp** command statement with an **access-list** command statement to permit outbound FTP traffic, as shown in the following example:

```
fixup protocol ftp strict ftp
access-list acl_in permit tcp any any eq ftp
access-group acl_in in interface inside
nat (inside) 1 0 0
global (outside) 1 209.165.201.5 netmask 255.255.255.224
```

6. IP addresses in the pool of global addresses specified with the **global** command require reverse DNS entries to ensure that all external network addresses are accessible through the PIX Firewall. To create reverse DNS mappings, use a DNS PTR record in the address-to-name mapping file for each global address. For more information on DNS, refer to *DNS and BIND*, by Paul Albitz and Cricket Liu, O'Reilly & Associates, Inc., ISBN 1-56592-010-4. Without the PTR entries, sites can experience slow or intermittent Internet connectivity and FTP requests that consistently fail. For example, if a global IP address is 209.165.201.1 and the domain for the PIX Firewall is pix.example.com, the PTR record would be:

```
1.201.165.209.in-addr.arpa. IN PTR pix.example.com.
```

7. A DNS server on a higher level security interface needing to get updates from a root name server on the outside interface cannot use PAT (Port Address Translation). Instead, a **static** command statement must be added to map the DNS server to a global address on the outside interface.

For example, PAT is enabled with these commands:

```
nat (inside) 1 192.168.1.0 255.255.255.0
global (inside) 1 209.165.202.128 netmask 255.255.255.224
```

However, a DNS server on the inside at IP address 192.168.1.5 cannot correctly reach the root name server on the outside at IP address 209.165.202.130.

To ensure that the inside DNS server can access the root name server, insert the following **static** command statement:

```
static (inside,outside) 209.165.202.129 192.168.1.5
```

The global address 209.165.202.129 provides a translated address for the inside server at IP address 192.168.1.5.

- The following example enables PAT using the IP address at the outside interface in global configuration mode:

```
ip address outside 192.150.49.1
nat (inside) 1 0 0
global (outside) 1 interface
```

The interface IP address used for PAT is the address associated with the interface when the xlate is created. This is important for configuring DHCP, allowing for the DHCP retrieved address to be used for PAT.

When PAT is enabled on an interface, there should be no termination of TCP, UDP, and ICMP services. These services allow for termination at the PIX Firewall's outside interface.

- To specify PAT using the IP address of an interface, specify the **interface** keyword.

**global** [(*int\_name*)] *nat\_id* *address* | **interface**

The following example enables PAT using the IP address at the outside interface in global configuration mode:

```
ip address outside 192.150.49.1
nat (inside) 1 0 0
global (outside) 1 interface
```

The interface IP address used for PAT is the address associated with the interface when the xlate (translation slot) is created. This is important for configuring DHCP, allowing for the DHCP retrieved address to be used for PAT.

When PAT is enabled on an interface, there should be no loss of TCP, UDP, and ICMP services. These services allow for termination at the PIX Firewall unit's outside interface.

- To track usage among different subnets, you can specify multiple PATs using the following supported configurations:

The following example maps hosts on the internal network 10.1.0.0/16 to global address 192.168.1.1 and hosts on the internal network 10.1.1.1/16 to global address 209.165.200.225 in global configuration mode.

```
nat (inside) 1 10.1.0.0 255.255.255.0
nat (inside) 2 10.1.1.1 255.255.255.0
global (outside) 1 192.168.1.1 netmask 255.255.255.0
global (outside) 2 209.165.200.225 netmask 255.255.255.224
```

The following example configures two port addresses for setting up PAT on hosts from the internal network 10.1.0.0/16 in global configuration mode.

```
nat (inside) 1 10.1.0.0 255.255.0.0
global (outside) 1 209.165.200.225 netmask 255.255.255.224
global (outside) 1 192.168.1.1 netmask 255.255.255.0
```

With this configuration, address 192.168.1.1 will only be used when the port pool from address 209.165.200.225 is at maximum capacity.

### Examples

The following example declares two global pool ranges and a PAT address. Then the **nat** command permits all inside users to start connections to the outside network:

```
global (outside) 1 209.165.201.1-209.165.201.10 netmask 255.255.255.224
global (outside) 1 209.165.201.12 netmask 255.255.255.224
Global 209.165.201.12 will be Port Address Translated
nat (inside) 1 0 0
clear xlate
```

The next example creates a global pool from two contiguous pieces of a Class C address and gives the perimeter hosts access to this pool of addresses to start connections on the outside interface:

```
global (outside) 1000 209.165.201.1-209.165.201.14 netmask 255.255.255.240
global (outside) 1000 209.165.201.17-209.165.201.30 netmask 255.255.255.240
nat (perimeter) 1000 0 0
```

# help

Display help information. (Unprivileged mode.)

**help**

**?**

## Usage Guidelines

The **help** or **?** command displays help information about all commands. You can view help for an individual command by entering the command name followed by a question mark or just the command name and pressing the **Enter** key.

If the **pager** command is enabled and when 24 lines display, the listing pauses, and the following prompt appears:

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

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

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

## Examples

The following example shows how you can display help information by following the command name with a question mark:

```
enable ?  
usage: enable password <pw> [encrypted]
```

Help information is available on the core commands (not the **show**, **no**, or **clear** commands) by entering **?** at the command prompt:

```
?  
aaa          Enable, disable, or view TACACS+ or RADIUS  
             user authentication, authorization and accounting  
...
```

# hostname

Change the host name in the PIX Firewall command line prompt. (Configuration mode.)

**hostname** *newname*

## Syntax Description

*newname* New host name for the PIX Firewall prompt. This name can be up to 16 alphanumeric characters and mixed case.

## Usage Guidelines

The **hostname** command changes the host name label on prompts. The default host name is pixfirewall.



### Note

---

The change of the host name causes the change of the fully qualified domain name. Once the fully qualified domain name is changed, delete the RSA key pairs with the **ca zeroize rsa** command and delete related certificates with the **no ca identity ca\_nickname** command.

---

## Examples

The following example shows how to change a host name:

```
pixfirewall(config)# hostname spinner
spinner(config)# hostname pixfirewall
pixfirewall(config)#
```

# icmp

Enable or disable ping to an interface. (Configuration mode.)

**icmp permit** | **deny** [**host**] *src\_addr* [*src\_mask*] [*type*] *int\_name*

**no icmp permit** | **deny** [**host**] *src\_addr* [*src\_mask*] [*type*] *int\_name*

**clear icmp**

**show icmp**

## Syntax Description

**permit** | **deny** Permit or deny the ability to ping a PIX Firewall interface.

*src\_addr* Address that is either permitted or denied ability to ping an interface. Use **host** *src\_addr* to specify a single host.

*src\_mask* Network mask. Specify if a network address is specified.

*type* ICMP message type as described in Table 5-7.

*int\_name* Interface name that can be pinged.

## Usage Guidelines

Enable or disable ping to an interface. With ping to an interface disabled, the PIX Firewall cannot be detected on the network. The new **icmp** command implements this feature. This feature is also referred to as configurable proxy ping.

To use the **icmp** command, configure an **access-list** command statement that permits or denies ICMP traffic that terminates at the PIX Firewall unit.

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

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

The syslog message is as follows:

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

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

## ICMP Message Types

Table 5-7 lists possible ICMP type values.

**Table 5-7 ICMP Type Literals**

ICMP Type	Literal
0	echo-reply
3	unreachable
4	source-quench
5	redirect
6	alternate-address
8	echo
9	router-advertisement
10	router-solicitation
11	time-exceeded
12	parameter-problem
13	timestamp-reply
14	timestamp-request
15	information-request
16	information-reply
17	mask-request
18	mask-reply
31	conversion-error
32	mobile-redirect

### Example

1. Deny all ping requests and permit all unreachable messages at the outside interface:

```
icmp deny any echo-reply outside
icmp permit any unreachable outside
```

2. Permit host 172.16.2.15 or hosts on subnet 172.22.1.0/16 to ping the outside interface:

```
icmp permit host 172.16.2.15 echo-reply outside
icmp permit 171.22.1.0 255.255.255.0 echo-reply outside
icmp permit any unreachable outside
```

# interface

Identify network interface speed and duplex. (Configuration mode.)

**interface** *hardware\_id* [*hardware\_speed*] [**shutdown**]

**clear interface**

**show interface**

## Syntax Description

<i>hardware_id</i>	Identifies the network interface type. Possible values are <b>ethernet0</b> , <b>ethernet1</b> to <b>ethernetn</b> , <b>gb-ethernetn</b> , <b>fddi0</b> or <b>fddi1</b> , <b>token-ring0</b> , <b>token-ring1</b> , to <b>token-ringn</b> , depending on how many network interfaces are in the firewall.
<i>hardware_speed</i>	Network interface speed (optional). Do not specify a <i>hardware_speed</i> for a FDDI interface.  Possible Ethernet values are: <b>10baset</b> —Set for 10 Mbps Ethernet half duplex communication. <b>10full</b> —Set for 10 Mbps Ethernet full duplex communication. <b>100basetx</b> —Set for 100 Mbps Ethernet half duplex communication. <b>100full</b> —Set for 100 Mbps Ethernet full duplex communication. <b>1000sxfull</b> —Set for 1000 Mbps Gigabit Ethernet full duplex operation. <b>1000basesx</b> —Set for 1000 Mbps Gigabit Ethernet half duplex operation. <b>1000auto</b> —Set for 1000 Mbps Gigabit Ethernet to auto-negotiate full or half duplex. Cisco recommends that you do not use this option to maintain compatibility with switches and other devices in your network. <b>au</b> <b>i</b> —Set 10 for Mbps Ethernet half duplex communication with an AUI cable interface. <b>auto</b> —Set Ethernet speed automatically. The <b>auto</b> keyword can only be used with the Intel 10/100 automatic speed sensing network interface card. Cisco recommends that you do not use this option to maintain compatibility with switches and other devices in your network. <b>bnc</b> —Set for 10 Mbps Ethernet half duplex communication with a BNC cable interface.  Possible Token Ring values are: <b>4mbps</b> —4 Mbps data transfer speed. You can specify this as <b>4</b> . <b>16mbps</b> —(default) 16 Mbps data transfer speed. You can specify this as <b>16</b> .
<b>shutdown</b>	Disable an interface.

### Usage Guidelines

The **interface** command identifies the speed and duplex settings of the network interface boards. Use **show interface** to view information about the interface. The **show interface** command displays the packet drop count of Unicast RPF for each interface. This value appears as the “unicast rpf drops” counter.

The **clear interface** command clears all interface statistics except the number of input bytes. This command no longer shuts down all system interfaces. The **clear interface** command works with all interface types except gigabit Ethernet. The **clear interface** command also clears the packet drop count of Unicast RPF for all interfaces.

The **shutdown** option lets you disable an interface. When you first install PIX Firewall, all interfaces are shut down by default. You must explicitly enable an interface by entering the command without the **shutdown** option. If the **shutdown** option does not exist in the command, packets are passed by the driver to and from the card.

If the **shutdown** option does exist, packets are dropped in either direction. Inserting a new card defaults to the default interface command containing the **shutdown** option. (That is, if you add a new card and then enter the **write memory** command, the **shutdown** option is saved into Flash memory for the interface.) When upgrading from a previous version to the current version, interfaces are enabled.

The configuration of the interface affects buffer allocation (the PIX Firewall will allocate more buffers for higher line speeds). Buffer allocation can be checked with the **show blocks** command.




---

**Note** For failover, set the Stateful Failover dedicated interface to 100 Mbps full duplex using the **100full** option to the **interface** command.

---




---

**Note** The **show interface** command reports “line protocol down” for BNC cable connections and for 3Com cards.

---




---

**Note** Even though the default is to set automatic speed sensing for the interfaces with the **interface hardware\_id auto** command, Cisco recommends that you specify the speed of the network interfaces; for example, **10baset** or **100basetx**. This lets PIX Firewall operate in network environments that may include switches or other devices that do not handle auto sensing correctly.

---

### Usage Notes

1. When you use the **interface token-ring** command, also use the **mtu** command to set the block size depending on the interface speed.
2. After changing an **interface** command, use the **clear xlate** command.

### show interface Notes

The **show interface** command lets you view network interface information for both Ethernet and Token Ring, depending on which is installed in your PIX Firewall. This is one of the first commands you should use when establishing network connectivity after installing a PIX Firewall.

The information in the **show interface** display is as follows:

- The ethernet, fddi, or token-ring interface strings indicate that you have used the **interface** command to configure the interface. The statement indicates either outside or inside and whether the interface is available (“up”) or not available (“down”).

- “line protocol up” means a working cable is plugged into the network interface. If the message is “line protocol down,” either the cable is incorrect or not plugged into the interface connector.
- Network interface type.
- Interrupt vector. It is acceptable for interface cards to have the same interrupts because PIX Firewall uses interrupts to get Token Ring information, but polls Ethernet cards.
- MAC address. Intel cards start with “i” and 3Com cards with “3c.”
- MTU (maximum transmission unit): the size in bytes that data can best be sent over the network.
- “*nn* packets input” indicates that packets are being received in the firewall.
- “*nn* packets output” indicates that packets are being sent from the firewall.
- Line duplex status: half duplex indicates that the network interface switches back and forth between sending and receiving information; full duplex indicates that the network interface can send or receive information simultaneously.
- Line speed: **10baset** is listed as 10,000 Kbit; **100basetx** is listed as 100,000 Kbit.
- Interface problems:
  - no buffer, the PIX Firewall is out of memory or slowed down due to heavy traffic and cannot keep up with the received data.
  - runts are packets with less information than expected.
  - giants are packets with more information than expected.
  - input errors.
  - CRC (cyclic redundancy check) are packets that contain corrupted data (checksum error).
  - frame errors are framing errors.
  - overruns occur when the network interface card is overwhelmed and cannot buffer received information before more needs to be sent.
  - ignored and aborted errors are provided for future use, but are not currently checked; the PIX Firewall does not ignore or abort frames.
  - underruns occur when the PIX Firewall is overwhelmed and cannot get data fast enough to the network interface card.
  - unicast rpf drops—when packets sent to a single network destination using reverse path forwarding are dropped.
  - output errors—(maximum collisions). The number of frames not transmitted because the configured maximum number of collisions was exceeded. This counter should only increment during heavy network traffic.
  - collisions—(single and multiple collisions). The number of messages retransmitted due to an Ethernet collision. This usually occurs on an overextended LAN (Ethernet or transceiver cable too long, more than two repeaters between stations, or too many cascaded multiport transceivers). A packet that collides is counted only once by the output packets.
  - interface resets—the number of times an interface has been reset. If an interface is unable to transmit for three seconds, PIX Firewall resets the interface to restart transmission. During this interval, connection state is maintained. An interface reset can also happen when an interface is looped back or shut down.
  - babbles—unused. (“babble” means that the transmitter has been on the interface longer than the time taken to transmit the largest frame.)

- late collisions—the number of frames that were not transmitted because a collision occurred outside the normal collision window. A late collision is a collision that is detected late in the transmission of the packet. Normally, these should never happen. When two Ethernet hosts try to talk at once, they should collide early in the packet and both back off, or the second host should see that the first one is talking and wait.

If you get a late collision, a device is jumping in and trying to send on the Ethernet while the PIX Firewall is partly finished sending the packet. The PIX Firewall does not resend the packet, because it may have freed the buffers that held the first part of the packet.

This is not a real problem because networking protocols are designed to cope with collisions by resending packets. However, late collisions indicate a problem exists in your network. Common problems are large repeated networks and Ethernet networks running beyond the specification.

- deferred—the number of frames that were deferred before transmission due to activity on the link.
- lost carrier—the number of times the carrier signal was lost during transmission.
- no carrier—unused.
- Gigabit interface cards do not provide information for the extended **show interface** command counters introduced in version 5.0(3).
- The **show interface** command has been enhanced to include eight additional status counters. The new counters are only valid for Ethernet interfaces. The following example shows the new output:

```
show interface
interface ethernet0 "outside" is up, line protocol is up
Hardware is i82559 ethernet, address is 00aa.0000.003b
IP address 209.165.201.7, subnet mask 255.255.255.224
MTU 1500 bytes, BW 100000 Kbit half duplex
  1184342 packets input, 1222298001 bytes, 0 no buffer
  Received 26 broadcasts, 27 runts, 0 giants
  4 input errors, 0 CRC, 4 frame, 0 overrun, 0 ignored, 0 abort
  1310091 packets output, 547097270 bytes, 0 underruns, 0 unicast rpf drops
  0 output errors, 28075 collisions, 0 interface resets
  0 babbles, 0 late collisions, 117573 deferred
  0 lost carrier, 0 no carrier
...
```

The counters in the last three lines are as follows:

- output errors—(maximum collisions). The number of frames not transmitted because the configured maximum number of collisions was exceeded. This counter should only increment during heavy network traffic.
- collisions—(single and multiple collisions). The number of messages retransmitted due to an Ethernet collision. This usually occurs on an overextended LAN (Ethernet or transceiver cable too long, more than two repeaters between stations, or too many cascaded multiport transceivers). A packet that collides is counted only once by the output packets.
- interface resets—the number of times an interface has been reset. If an interface is unable to transmit for three seconds, PIX Firewall resets the interface to restart transmission. During this interval, connection state is maintained. An interface reset can also happen when an interface is looped back or shut down.
- babbles—unused. (“babble” means that the transmitter has been on the interface longer than the time taken to transmit the largest frame.)

- late collisions—the number of frames that were not transmitted because a collision occurred outside the normal collision window. A late collision is a collision that is detected late in the transmission of the packet. Normally, these should never happen. When two Ethernet hosts try to talk at once, they should collide early in the packet and both back off, or the second host should see that the first one is talking and wait.
- If you get a late collision, a device is jumping in and trying to send the packet on the Ethernet while the PIX Firewall is partly finished sending the packet. The PIX Firewall does not resend the packet, because it may have freed the buffers that held the first part of the packet. This is not a real problem because networking protocols are designed to cope with collisions by resending packets. However, late collisions indicate a problem exists in your network. Common problems are large repeated networks and Ethernet networks running beyond the specification.
- deferred—the number of frames that were deferred before transmission due to activity on the link.
- lost carrier—the number of times the carrier signal was lost during transmission.
- no carrier—unused.

### Examples

The following example assigns names to each interface, enables auto detection for the interface parameters, and then shows interface activity:

```

show interface
interface ethernet0 "outside" is up, line protocol is up
  Hardware is i82557 ethernet, irq 10, address is 0060.7380.2f16
  IP address 209.165.201.1, subnet mask 255.255.255.224
  MTU 1500 bytes, BW 100000 Kbit half duplex
    0 packets input, 0 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    1 packets output, 0 bytes, 0 underruns, 0 unicast rpf drops
    0 output errors, 28075 collisions, 0 interface resets
    0 babbles, 0 late collisions, 117573 deferred
    0 lost carrier, 0 no carrier
interface token-ring0 "inside" is up, line protocol is up
  Hardware is o3137 token-ring, irq 9, address is 0000.8326.72c6
  IP address 10.0.0.1, subnet mask 255.0.0.0
  MTU 8192 bytes, BW 16000 Kbit, Ring-speed: 16Mbps
    116 packets input, 27099 bytes, 0 no buffer
    Received 116 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 116 frame, 0 overrun, 0 ignored, 0 abort
    3 packets output, 150 bytes, 0 underruns, 0 unicast rpf drops
    0 output errors, 28075 collisions, 0 interface resets
    0 babbles, 0 late collisions, 117573 deferred
    0 lost carrier, 0 no carrier
interface ethernet1 "DMZ" is up, line protocol is up
  Hardware is i82557 ethernet, irq 9, address is 00a0.c95d.0282
  IP address 127.0.0.1, subnet mask 255.255.255.0
  MTU 1500 bytes, BW 10000 Kbit half duplex
    0 packets input, 0 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    0 packets output, 0 bytes, 0 underruns, 0 unicast rpf drops
    0 output errors, 28075 collisions, 0 interface resets
    0 babbles, 0 late collisions, 117573 deferred
    0 lost carrier, 0 no carrier

```

# ip address

Identify addresses for network interfaces. (Configuration mode.)

```
ip address if_name ip_address [netmask]
```

```
ip address if_name dhcp [setroute]
```

```
show ip address if_name [dhcp]
```

```
show ip
```

```
clear ip
```

## Syntax Description

<i>if_name</i>	The internal or external interface name designated by the <b>nameif</b> command.
<i>ip_address</i>	PIX Firewall unit's network interface IP address.
<i>netmask</i>	Network mask of <i>ip_address</i> .
<b>dhcp</b>	Enable the DHCP client feature on the specified interface.
<b>setroute</b>	This option tells the PIX Firewall to set the default route using the default gateway parameter the DHCP server returns.

## Usage Guidelines

The **ip address** command lets you assign an IP address to each interface. Use the **show ip** command to view which addresses are assigned to the network interfaces. If you make a mistake while entering this command, re-enter the command with the correct information. The **clear ip** command resets all interface IP addresses to 127.0.0.1. The **clear ip** command does not affect the **ip local pool** or **ip verify reverse-route** commands.



Note

---

The **clear ip** command stops all traffic through the PIX Firewall unit.

---

After changing an **ip address** command, use the **clear xlate** command.



Note

---

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

---

**Note**

Do not set the netmask to all 255s, such as 255.255.255.255. This stops access on the interface. Instead, use a network address of 255.255.255.0 for Class C addresses, 255.255.0.0 for Class B addresses, or 255.0.0.0 for Class A addresses.

The default address for an interface is 127.0.0.1.

PIX Firewall configurations using failover require a separate IP address for each network interface on the Standby unit. The system IP address is the address of the Active unit. When the **show ip** command is executed on the Active unit, the current IP address is the same as the system IP address. When the **show ip** command is executed on the Standby unit, the system IP address is the failover IP address configured for the Standby unit.

The **ip address dhcp** command enables the DHCP client feature within the PIX Firewall. This command allows the PIX Firewall to be a DHCP client to a DHCP server that provides configuration parameters to the client. In this case, the configuration parameters the DHCP server provides is an IP address and a subnet mask to the interface on which the DHCP client feature is enabled. The optional **setroute** argument tells the PIX Firewall to set the default route using the default gateway parameter the DHCP server returns. If the **setroute** argument is configured, the **show route** command output shows the default route as being set by a DHCP server. To reset the interface and delete the DHCP lease from PIX Firewall, use the **clear ip** command. To clear the DHCP default route, use the **clear route static** command.

**Note**

Do not configure the PIX Firewall with a default route when using the **setroute** argument of the **ip address dhcp** command.

The **show ip address dhcp** command displays detailed information about the DHCP lease.

See “DHCP Client” within the Chapter 3, “Advanced Configurations” for more information about the DHCP client feature.

**Examples**

The following is sample output for the **show ip** command:

```
show ip
System IP Addresses:
  ip address outside 209.165.201.2 255.255.255.224
  ip address inside 192.168.2.1 255.255.255.0
  ip address perimeter 192.168.70.3 255.255.255.0
Current IP Addresses:
  ip address outside 209.165.201.2 255.255.255.224
  ip address inside 192.168.2.1 255.255.255.0
  ip address perimeter 192.168.70.3 255.255.255.0
```

The Current IP Addresses are the same as the System IP Addresses on the failover Active unit. When the Primary unit fails, the Current IP Addresses become those of the Standby unit.

The following is sample output for the **show ip address dhcp** command:

```
show ip address outside dhcp
```

```
Temp IP Addr:209.165.201.57 for peer on interface:outside  
Temp sub net mask:255.255.255.224  
DHCP Lease server:209.165.200.225, state:3 Bound  
DHCP Transaction id:0x4123  
Lease:259200 secs, Renewal:129600 secs, Rebind:226800 secs  
Temp default-gateway addr:209.165.201.1  
Next timer fires after:111797 secs  
Retry count:0, Client-ID:cisco-0000.0000.0000-outside
```

# ip audit

Configure IDS signature use. (Configuration mode.)

**ip audit attack** [action [alarm] [drop] [reset]]

**no ip audit attack**

**show ip audit attack**

**ip audit info** [action [alarm] [drop] [reset]]

**no ip audit info**

**show ip audit info**

**ip audit interface** *if\_name* *audit\_name*

**no ip audit interface** [*if\_name*]

**show ip audit interface**

**ip audit name** *audit\_name* **attack** [action [alarm] [drop] [reset]]

**no ip audit name** *audit\_name* [**attack**]

**show ip audit name** [**name** [info | attack]]

**ip audit name** *audit\_name* **info** [action [alarm] [drop] [reset]]

**no ip audit name** *audit\_name* [**info**]

**show ip audit name**

**ip audit signature** *signature\_number* **disable**

**no ip audit signature** *signature\_number*

**show ip audit signature** [*signature\_number*]

### Syntax Description

<b>audit attack</b>	Specify the default actions to be taken for attack signatures.
<b>audit info</b>	Specify the default actions to be taken for informational signatures.
<b>audit interface</b>	Apply an audit specification or policy (via the <b>ip audit name</b> command) to an interface.
<b>audit name</b>	Specify informational signatures, except those disabled or excluded by the <b>ip audit signature</b> command, as part of the policy.
<b>audit signature</b>	Specify which messages to display, attach a global policy to a signature, and disable or exclude a signature from auditing.
<b>action actions</b>	The <b>alarm</b> option indicates that when a signature match is detected in a packet, PIX Firewall reports the event to all configured syslog servers. The <b>drop</b> option drops the offending packet. The <b>reset</b> option drops the offending packet and closes the connection if it is part of an active connection. The default is <b>alarm</b> .
<i>audit_name</i>	Audit policy name viewed with the <b>show ip audit name</b> command.
<i>signature_number</i>	IDS signature number.

### Usage Guidelines

Cisco Secure Intrusion Detection System (Cisco Secure IDS) is an IP-only feature that provides some level of flexibility for the user to customize the amount of traffic that needs to be audited and logged.

The Cisco Secure IDS features provide the following:

- Traffic auditing. Application level signatures will only be audited as part of an active session.
- Apply the audit to an interface.
- Support different audit policies. Traffic matching a signature triggers a range of configurable actions.
- Disable the signature audit.
- Enable IDS and still disable actions of a signature class (informational, attack).

Auditing is performed by looking at the IP packets as they arrive at an input interface, if a packet triggers a signature and the configured action does not drop the packet, then the same packet can trigger other signatures.

PIX Firewall supports both inbound and outbound auditing.

For a complete list of supported Cisco Secure IDS signatures, their wording, and whether they are attack or informational messages, refer to *System Log Messages for the Cisco Secure PIX Firewall Version 5.2*. You can view this document online at the following site:

[http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix\\_v52/syslog/index.htm](http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_v52/syslog/index.htm)

Refer to the *Cisco Secure Intrusion Detection System Version 2.2.1 User Guide* for detailed information on each signature. You can view the “NSDB and Signatures” chapter of this guide at the following site:

<http://www.cisco.com/univercd/cc/td/doc/product/iaabu/csids/csids1/csidsug/sigs.htm>

The **ip audit** commands are described in the sections that follow.

#### ip audit attack

The **ip audit attack** **[action [alarm] [drop] [reset]]** command specifies the default actions to be taken for attack signatures. An audit policy (audit rule) defines the attributes for all signatures that can be applied to an interface along with a set of actions. Using an audit policy may limit the traffic that is audited or specify actions to be taken when the signature matches. Each audit policy is identified by a name and can be defined for informational or attack signatures. Each interface can have two policies; one for informational signatures and one for attack signatures. If a policy is defined without actions, then the configured default actions will take effect. Each policy requires a different name.

The **no ip audit attack** command resets the action to be taken for attack signatures to the default action. The **show ip audit attack** command displays the default attack actions.

#### ip audit info

The **ip audit info** **[action [alarm] [drop] [reset]]** command specifies the default action to be taken for signatures classified as informational signatures.

The **no ip audit info** command sets the action to be taken for signatures classified as informational and reconnaissance to the default action. The **show ip audit info** displays the default informational actions.

To cancel event reactions, specify the **ip audit info** command without an **action** option.

#### ip audit interface

The **ip audit interface** *if\_name* *audit\_name* command applies an audit specification or policy (via the **ip audit name** command) to an interface. The **no ip audit interface** [*if\_name*] command removes a policy from an interface. The **show ip audit interface** command displays the interface configuration.

#### ip audit name

The **ip audit name** *audit\_name* **info** **[action [alarm] [drop] [reset]]** command specifies the informational signatures except those disabled or excluded by the **ip audit signature** command that are considered part of the policy. The **no ip audit name** *audit\_name* [**info**] command removes the audit policy *audit\_name*. The **show ip audit name** **[name [info|attack]]** command displays all audit policies or specific policies referenced by name and possibly type.

#### ip audit signature

The **ip audit signature** *signature\_number* **disable** command specifies which messages to display, attaches a global policy to a signature, and disables or excludes a signature from auditing. The **no ip audit signature** *signature\_number* command removes the policy from a signature. Used to reenable a signature. The **show ip audit signature** [*signature\_number*] displays disabled signatures.

### Supported IDS Signatures

PIX Firewall lists the following single-packet IDS signature messages: 1000-1006, 1101, 1103, 2000-2012, 2150, 2151, 2154, 3040-3042, 4050-4052, 6050-6053, 6100-6103, 6150-6155, 6175, 6180, 6190, and 8000. All signature messages are not supported by PIX Firewall in this release. IDS syslog messages all start with %PIX-4-4000nn and have the following format:

```
%PIX-4-4000nn IDS:sig_num sig_msg from faddr to laddr on interface int_name
```

For example:

```
%PIX-4-400013 IDS:2003 ICMP redirect from 10.4.1.2 to 10.2.1.1 on interface dmz
%PIX-4-400032 IDS:4051 UDP Snork attack from 10.1.1.1 to 192.168.1.1 on interface outside
```

Options:

*sig\_num* The signature number.

*sig\_msg* The signature message—approximately the same as the Cisco Secure IDS signature message.

*faddr* The IP address of the foreign host initiating the attack. (“Foreign” is relative; attacks can be perpetrated either from outside to an inside host, or from the inside to an outside host.)

*laddr* The IP address of the local host to which the attack is directed. (“Local” is relative; attacks can be perpetrated either from outside to an inside host, or from the inside to an outside host.)

*int\_name* The name of the interface on which the signature originated.

### Examples

Disable signature 6102 globally:

```
ip audit signature 6102 disable
```

Specify default informational actions:

```
ip audit name attack1 info
```

Specify an attack policy:

```
ip audit name attack2 attack action alarm drop reset
```

Apply a policy to an interface:

```
ip audit interface outside attack1
ip audit interface inside attack2
```

# ip local pool

Identify addresses for a local pool. (Configuration mode)

```
ip local pool pool_name pool_start-address[-pool_end-address]  
no ip local pool pool_name pool_start-address[-pool_end-address]  
show ip local pool pool_name ip_address[-ip_address]
```

## Syntax Description

*pool\_name* Local pool name.

*pool\_start\_address* Local pool IP address range.

*pool\_end\_address*

## Usage Guidelines

The **ip local pool** command lets you create a pool of local addresses to be used for assigning dynamic ip addresses to remote VPN clients. The address range of this pool of local addresses must not overlap with any command statement that lets you specify an IP address. To delete an address pool, use the **no ip local pool** command. Use the **show ip local pool** command to view usage information about the pool of local addresses.

When a pool of addresses set by the **ip local pool** command is empty, the following syslog message appears:

```
%PIX-4-404101: ISAKMP: Failed to allocate address for client from pool poolname
```

To reference this pool of local addresses, use the **isakmp client configuration address-pool** command. Refer to the *IPSec User Guide for the Cisco Secure PIX Firewall Version 5.2* for information on the **isakmp** command.

## Examples

The following example creates a pool of IP addresses and then displays the pool contents:

```
ip local pool mypool 10.0.0.10-10.0.0.20  
show ip local pool mypool
```

```
Pool          Begin          End            Free    In use  
mypool        10.0.0.10     10.0.0.20     11      0
```

```
Available Addresses:  
10.0.0.10  
10.0.0.11  
10.0.0.12  
10.0.0.13  
10.0.0.14  
10.0.0.15  
10.0.0.16  
10.0.0.17  
10.0.0.18  
10.0.0.19  
10.0.0.20
```

# ip verify reverse-path

Implement unicast RPF IP spoofing protection. (Configuration mode.)

```
ip verify reverse-path interface int_name
```

```
no ip verify reverse-path interface int_name
```

```
show ip verify [reverse-path [interface int_name]]
```

```
clear ip verify [reverse-path [interface int_name]]
```

## Syntax Description

*int\_name* Name of an interface you want to protect from a DoS attack.

## Usage Guidelines

The **ip verify reverse-path** command lets you specify which interfaces to protect from an IP spoofing attack using network ingress and egress filtering, which is described in RFC 2267. This command is disabled by default and provides unicast RPF (Reverse Path Forwarding) functionality for the PIX Firewall. The **show ip verify** command lists the **ip verify** commands in the configuration. The **clear ip verify** command removes **ip verify** commands from the configuration. Unicast RPF is a unidirectional input function that screens inbound packets arriving on an interface. Outbound packets are not screened.

Due to the danger of IP spoofing in the IP protocol, measures need to be taken to reduce this risk when possible. Unicast RPF (Reverse Path Forwarding), or reverse route lookups, prevents such manipulation under certain circumstances.



### Note

---

The **ip verify reverse-path** command depends on the existence of a default route statement in the configuration for the outside interface that has 0.0.0.0 0.0.0.0 in the **route** command statement for the IP address and network mask.

---

The **ip verify reverse-path** command provides both ingress and egress filtering. Ingress filtering checks inbound packets for IP source address integrity, and is limited to addresses for networks in the enforcing entity's local routing table. If the incoming packet does not have a source address represented by a route, then it is impossible to know whether the packet has arrived on the best possible path back to its origin. This is often the case when routing entities cannot maintain routes for every network.

Egress filtering verifies that packets destined for hosts outside the managed domain have IP source addresses verifiable by routes in the enforcing entity's local routing table. If an exiting packet does not arrive on the best return path back to the originator, then the packet is dropped and the activity logged. Egress filtering prevents internal users from launching attacks using IP source addresses outside of the local domain because most attacks use IP spoofing to hide the identity of the attacking host. Egress filtering makes the task of tracing the origin of an attack much easier. When employed, egress filtering enforces what IP source addresses are obtained from a valid pool of network addresses. Addresses are kept local to the enforcing entity and are therefore easily traceable.

Unicast RPF is implemented as follows:

- ICMP packets have no session so each packet is checked.
- UDP and TCP have sessions, so the initial packet requires a reverse route lookup. Subsequent packets arriving during the session are checked using an existing state maintained as part of the session. Non-initial packets are checked to ensure they arrived on the same interface used by the initial packet.



#### Note

Before using this command, add static **route** command statements for every network that can be accessed on the interfaces you wish to protect. Only enable this command if routing is fully specified. Otherwise, PIX Firewall will stop traffic on the interface you specify if routing is not in place.

Use the **show interface** command to view the number dropped packets, which appears in the “unicast rpf drops” counter.

#### Examples

The following example protects traffic between the inside and outside interfaces and provides **route** command statements for two networks 10.1.2.0 and 10.1.3.0 that connect to the inside interface via a hub:

```
ip address inside 10.1.1.1 255.255.0.0
route inside 10.1.2.0 255.255.0.0 10.1.1.1 1
route inside 10.1.3.0 255.255.0.0 10.1.1.1 1
ip verify reverse-path interface outside
ip verify reverse-path interface inside
```

The **ip verify reverse-path interface outside** command statement protects the outside interface from network ingress attacks from the Internet, whereas the **ip verify reverse-path interface inside** command statement protects the inside interface from network egress attacks from users on the internal network.

# kill

Terminate a Telnet session. (Privileged mode.)

```
kill telnet_id
```

## Syntax Description

*telnet\_id*      Telnet session ID.

## Usage Guidelines

The **kill** command terminates a Telnet session. Use the **who** command to view the Telnet session ID value. When you kill a Telnet session, the PIX Firewall lets any active commands terminate and then drops the connection without warning the user. The **kill** command does not affect PIX Firewall Manager sessions.

See also: **show who**, **telnet**.

## Examples

The following is sample output from the **show who** command, which is used to list the active Telnet sessions, and the use of the **kill** command to end Telnet session 2:

```
show who  
2: From 10.10.54.0  
kill 2
```

# local-host (clear and show)

View local host network states. (Privileged mode (**show**), configuration mode (**clear**)).

```
clear local-host [ip_address]
```

```
show local-host [ip_address]
```

## Syntax Description

*ip\_address* Local host IP address.

## Usage Guidelines

The **show local-host** command lets you view the network states of local hosts. Local hosts are any hosts on the same subnet as an internal PIX Firewall interface (not the outside interface). Hosts beyond the next hop routers are not affected by this command.

This command lets you show the translation and connection slots for the local hosts, or stop all traffic on these hosts. This command provides information for hosts configured with the **nat 0** command when normal translation and connection states may not apply.

Use the *ip\_address* option to limit the display to a single host. The **clear local-host** command clears the information displayed for the local host.



### Note

---

Clearing the network state of a local host stops all connections and xlates associated with the local hosts.

---

## Examples

The following is sample output from the **show local-host** command:

```
show local-host 10.1.1.15
local host: <10.1.1.15>, conn(s)/limit = 2/0, embryonic(s)/limit = 0/0
  Xlate(s):
    PAT Global 172.16.3.200(1024) Local 10.1.1.15(55812)
    PAT Global 172.16.3.200(1025) Local 10.1.1.15(56836)
    PAT Global 172.16.3.200(1026) Local 10.1.1.15(57092)
    PAT Global 172.16.3.200(1027) Local 10.1.1.15(56324)
    PAT Global 172.16.3.200(1028) Local 10.1.1.15(7104)
  Conn(s):
    TCP out 192.150.49.10:23 in 10.1.1.15:1246 idle 0:00:20 Bytes 449 flags UIO
    TCP out 192.150.49.10:21 in 10.1.1.15:1247 idle 0:00:10 Bytes 359 flags UIO
```

The xlate describes the translation slot information and the Conn is the connection state information.

The next example shows how the clear local-host command clears the local host information:

```
clear local-host 10.1.1.15
show local-host 10.1.1.15
```

Once the information is cleared, nothing more displays until the hosts reestablish their connections, which were stopped by the **clear local-host** command, and more data is produced.

# logging

Enable or disable syslog and SNMP logging. (Configuration mode.)

**logging on**

**no logging on**

**logging buffered** *level*

**no logging buffered**

**logging console** *level*

**no logging console**

**logging facility** *facility*

**no logging facility** *facility*

**logging history** *level*

**no logging history** *level*

**logging host** [*in\_if\_name*] *ip\_address* [*protocol/port*]

**no logging host** [*in\_if\_name*] *ip\_address*

**logging message** *syslog\_id*

**no logging message** *syslog\_id*

**clear logging disabled**

**show logging disabled**

**logging monitor** *level*

**no logging monitor** *level*

**logging queue** *messages*

**no logging queue** *messages*

**show logging queue**

**logging standby**

**no logging standby**

**logging timestamp**

**no logging timestamp**

**logging trap** *level*

**no logging trap** *level*

**show logging**

**clear logging**

#### Syntax Description

- |                 |   |
|-----------------|---|
| <b>on</b>       | Start sending syslog messages to all output locations. Stop all logging with the <b>no logging on</b> command.  |
| <b>buffered</b> | Send syslog messages to an internal buffer that can be viewed with the <b>show logging</b> command. Use the <b>clear logging</b> command to clear the message buffer. New messages append to the end of the buffer.   |
| <i>level</i>    | Specify the syslog message level as a number or string. The <i>level</i> you specify means that you want that <i>level</i> and those less than the <i>level</i> . For example, if <i>level</i> is <b>3</b> , syslog displays <b>0</b> , <b>1</b> , <b>2</b> , and <b>3</b> messages. Possible number and string <i>level</i> values are: <ul style="list-style-type: none"> <li>• <b>0—emergencies</b>—System unusable messages</li> <li>• <b>1—alerts</b>—Take immediate action</li> <li>• <b>2—critical</b>—Critical condition</li> <li>• <b>3—errors</b>—Error message</li> <li>• <b>4—warnings</b>—Warning message</li> <li>• <b>5—notifications</b>—Normal but significant condition</li> <li>• <b>6—informational</b>—Information message</li> <li>• <b>7—debugging</b>—Debug messages and log FTP commands and WWW URLs</li> </ul> |

<b>console</b>	Specify that syslog messages appear on the PIX Firewall console as each message occurs. You can limit the types of messages that appear on the console with <i>level</i> . Cisco recommends that you do not use this command in production mode because its use degrades PIX Firewall performance.
<b>facility</b>	Specify the syslog facility. The default is 20.
<i>facility</i>	Eight facilities LOCAL0(16) through LOCAL7(23); the default is LOCAL4(20). Hosts file the messages based on the <i>facility</i> number in the message.
<b>history</b>	Set the SNMP message level for sending syslog traps.
<b>host</b>	Specify a syslog server that will receive the messages sent from the PIX Firewall. You can use multiple <b>logging host</b> commands to specify additional servers that would all receive the syslog messages. However a server can only be specified to receive either UDP or TCP, not both. PIX Firewall only sends TCP syslog messages to the PIX Firewall Syslog Server.
<i>in_if_name</i>	Interface on which the syslog server resides.
<i>ip_address</i>	Syslog server's IP address.
<i>protocol</i>	The protocol over which the syslog message is sent; either <b>tcp</b> or <b>udp</b> . PIX Firewall only sends TCP syslog messages to the PIX Firewall Syslog Server. You can only view the port and protocol values you previously entered by using the <b>write terminal</b> command and finding the command in the listing—the TCP protocol is listed as 6 and the UDP protocol is listed as 17.
<i>port</i>	The port from which the PIX Firewall sends either UDP or TCP syslog messages. This must be same port at which the syslog server. For the UDP port, the default is 514 and the allowable range for changing the value is 1025 through 65535. For the TCP port, the default is 1470, and the allowable range is 1025 through 65535.
<b>message</b>	Specify a message to be allowed. Use with the <b>no</b> command to suppress a message. Use the <b>clear logging disabled</b> command to reset the disallowed messages to the original set. Use the <b>show message disabled</b> command to list the suppressed messages you specified with the <b>no logging message</b> command. All syslog messages are permitted unless explicitly disallowed. The “PIX Startup begin” message cannot be blocked and neither can more than one message per command statement.
<i>syslog_id</i>	Specify a message number to disallow or allow. If a message is listed in syslog as %PIX-1-101001, use “101001” as the <i>syslog_id</i> . Refer to the <i>System Log Messages for the Cisco Secure PIX Firewall Version 5.2</i> guide for message numbers. You can view this document online at the following site:  <a href="http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_v52/syslog/index.htm">http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_v52/syslog/index.htm</a>
<b>disabled</b>	Clear or display suppressed messages. You can suppress messages with the <b>no logging message syslog_id</b> command.
<b>monitor</b>	Specify that syslog messages appear on Telnet sessions to the PIX Firewall console.

<b>queue messages</b>	Specify the number of syslog messages permitted to be queued before being processed. The messages parameter defaults to 512, 0 (zero) indicates unlimited, and the minimum is one message. The maximum number of messages is limited by available memory. Use the <b>show logging queue</b> command to determine the number of messages in the queue.
<b>standby</b>	Let the failover Standby unit also send syslog messages. This option is disabled by default. You can enable it to ensure that the Standby unit's syslog messages stay synchronized should failover occur. However, this option causes twice as much traffic on the syslog server. Disable with the <b>no logging standby</b> command.
<b>timestamp</b>	Specify that syslog messages sent to the syslog server should have a time stamp value on each message.
<b>trap</b>	Set logging level only for syslog messages.
<b>clear</b>	Clear the buffer for use with the <b>logging buffered</b> command.
<b>show</b>	List which logging options are enabled. If the <b>logging buffered</b> command is in use, the <b>show logging</b> command lists the current message buffer.

### Usage Guidelines

The **logging** command lets you enable or disable sending informational messages to the console, to a syslog server, or to an SNMP management station. Set the SNMP message level with the **logging history** command, and set the syslog message level with the **logging trap** command.

The **logging queue** command lets you specify the size of the syslog message queue, the messages waiting to be processed. When traffic is heavy, messages may be discarded. The **show logging queue** command lists the number of messages in the queue, the most number that have been in the queue, and the number of messages discarded because block memory was not available to process them. The **logging standby** command lets the failover Standby unit send syslog messages. This option is disabled by default. You can enable it to ensure that the Standby unit's syslog messages stay synchronized should failover occur. However, this option causes twice as much traffic on the syslog server. Disable with the **no logging standby** command.

You can also use this guide to get the message numbers that can be individually suppressed with the logging message command.

For more information on syslog and the use of the **logging** command, refer to “Step 16—Enable Syslog” in Chapter 2, “Configuring the PIX Firewall,” “IDS Syslog Messages” in Chapter 3, “Advanced Configurations,” and to *System Log Messages for the Cisco Secure PIX Firewall Version 5.2*. You can view this document online at the following site:

[http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix\\_v52/syslog/index.htm](http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_v52/syslog/index.htm)

### Important Notes

1. Do not use the **logging console** command when the PIX Firewall is in production mode because it degrades system performance. By default, this command is disabled. Instead, use the **logging buffered** command to start logging, the **show logging** command to view the messages, and the **clear logging** command to clear the buffer to make viewing the most current messages easier.
2. PIX Firewall provides more information in messages sent to a syslog server than at the console, but the console provides enough information to permit effective troubleshooting.
3. The **logging timestamp** command requires that the **clock** command be set.

4. The **no logging message** command cannot block the “%PIX-6-199002: PIX startup completed. Beginning operation.” syslog message.
5. The **aaa authentication enable console** command causes syslog messages to be sent (at syslog level 4) each time the configuration is changed from the serial console.

See also: **clear Commands, telnet, terminal**

### Examples

The following example shows how to start console logging and view the results:

```
logging buffered debugging
show logging
Syslog logging: enabled
  Timestamp logging: disabled
  Console logging: disabled
  Monitor logging: disabled
  Buffer logging: level debugging, 37 messages logged
  Trap logging: disabled
305001: Portmapped translation built for gaddr 209.165.201.5/0 laddr 192.168.1.2/256
...
```

The line of output starting with 305001 shows a translation to a PAT global through global address 209.165.201.5 from a host at 192.168.1.2. The “305001” identifies a syslog message for creating a translation through a PAT global. Refer to *System Log Messages for the Cisco Secure PIX Firewall Version 5.2* for more information on syslog messages. You can view this document online at the following site:

[http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix\\_v51/syslog/index.htm](http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_v51/syslog/index.htm)

The next example lists the output of the **logging queue** and **show logging queue** commands:

```
logging queue 0
show logging queue
Logging Queue length limit : Unlimited
Current 5 msg on queue, 3513 msgs most on queue, 1 msg discard.
```

In this example, the **logging queue** command is set to 0, which means you want an unlimited number of messages; in other words, all syslog messages, to be processed. The **show logging queue** command shows that 5 messages are queued, 3513 messages was the greatest number of messages in the queue at one time since the PIX Firewall was last booted, and that 1 message was discarded. Even though set for unlimited, should the amount of block memory be exhausted, messages can still be discarded.

# mtu

Specify the MTU (maximum transmission unit) for an interface. (Configuration mode.)

```
mtu if_name bytes
no mtu [if_name bytes]
show mtu
```

## Syntax Description

<i>if_name</i>	The internal or external network interface name.
<i>bytes</i>	The number of bytes in the MTU, in the range of 64 to 65,535 bytes. The value specified depends on the type of network connected to the interface.

## Usage Guidelines

The **mtu** command sets the size of data sent on a connection. Data larger than the MTU value is fragmented before being sent. The minimum value for *bytes* is 64 and the maximum is 65,535 bytes.

PIX Firewall supports the IP Path MTU Discovery mechanism, as defined in RFC 1191. IP Path MTU Discovery allows a host to dynamically discover and cope with differences in the maximum allowable maximum transmission unit (MTU) size of the various links along the path. Sometimes a PIX Firewall is unable to forward a datagram because it requires fragmentation (the packet is larger than the MTU you set for the interface), but the “don't fragment” (DF) bit is set. The network software sends a message to the sending host, alerting it to the problem. The host will have to fragment packets for the destination so that they fit the smallest packet size of all the links along the path.

For Ethernet interfaces, the default MTU is 1,500 bytes in a block, which is also the maximum. This value is sufficient for most applications, but you can pick a lower number if network conditions warrant it.

For Token Ring and FDDI, the default is 8,192 bytes.

The **no mtu** command resets the MTU block size to 1,500 for Ethernet interfaces and 8,192 for Token Ring. The **show mtu** command displays the current block size. The **show interface** command also shows the MTU value.

## Examples

The following example shows use of the **mtu** command for use with Token Ring and Ethernet:

```
interface token-ring0 16mbps
interface ethernet0 auto
mtu inside 8192
show mtu
mtu outside 1500
mtu inside 8192
```

# name/names

Associate a name with an IP address. (Configuration mode.)

```
name ip_address name  
no name [ip_address name]  
names  
no names  
clear names  
show names
```

## Syntax Description

*ip\_address* The IP address of the host being named.

*name* The name assigned to the IP address. Allowable characters are **a** to **z**, **A** to **Z**, **0** to **9**, a dash, and an underscore. The *name* cannot start with a number. If the name is over 16 characters long, the **name** command fails.

## Usage Guidelines

Use the **name** command to identify a host by a text name. The names you define become like a host table local to the PIX Firewall. Because there is no connection to DNS or /etc/hosts on UNIX servers, use of this command is a mixed blessing—it makes configurations much more readable but introduces another level of abstraction to administer; not only do you have to add and delete IP addresses to your configuration as you do now, but with this command, you need to ensure that the host names either match existing names or you have a map to list the differences.

The **names** command enables use of the **name** command to map text strings to IP addresses. The **clear names** and **no names** commands are the same and disable use of the **name** text strings. The **show names** command lists the **name** command statements in the configuration.

## Notes

1. You must first use the **names** command before using the **name** command. Use the **name** command immediately after the **names** command and before you use the write memory command.
2. To disable displaying **name** values, use **no names**.
3. Only one name can be associated with an IP address.
4. Both **name** and **names** command statements are saved in the configuration.
5. While the **name** command will let you assign a name to a network mask, no other PIX Firewall command requiring a mask will let you use the name as a mask value. For example, the following command is accepted:

```
name 255.255.255.0 class-C-mask
```

But, none of the commands in which a mask is required can process the “class-C-mask” as a accepted network mask.

### Examples

In the example that follows, the **names** command enables use of the **name** command. The **name** command substitutes **pix\_inside** for references to 192.168.42.3, and **pix\_outside** for 209.165.201.3. The **ip address** commands use these names while assigning IP addresses to the network interfaces. The **no names** command disables the **name** values from displaying. Subsequent use of the **names** command restores their display:

```
names
name 192.168.42.3 pix_inside
name 209.165.201.3 pix_outside
ip address inside pix_inside 255.255.255.0
ip address outside pix_outside 255.255.255.224
show ip address
inside ip address pix_inside mask 255.255.255.0
outside ip address pix_outside mask 255.255.255.224
no names
show ip address
inside ip address 192.168.42.3 mask 255.255.255.0
outside ip address 209.165.201.3 mask 255.255.255.224
names
show ip address
inside ip address pix_inside mask 255.255.255.0
outside ip address pix_outside mask 255.255.255.224
```

# nameif

Name interfaces and assign security level. (Configuration mode.)

```
nameif hardware_id if_name security_level
```

```
show nameif
```

## Syntax Description

<i>hardware_id</i>	<p>The hardware name for the network interface that specifies the interface's slot location on the PIX Firewall motherboard. Interface boards are numbered from the leftmost slot nearest the power supply as slot 0. The internal network interface must be in slot 1. The lowest <i>security_level</i> external interface board is in slot 0 and the next lowest <i>security_level</i> external interface board is in slot 2.</p> <p>Possible choices are <b>ethernet</b><i>n</i> for Ethernet or <b>token-ring</b><i>n</i> for Token Ring. These names can be abbreviated with any leading characters in the name; for example, <b>ether1</b>, <b>e2</b>, <b>token0</b>, or <b>t0</b>.</p>
<i>if_name</i>	<p>A name for the internal or external network interface of up to 48 characters in length. This name can be uppercase or lowercase. By default, PIX Firewall names the inside interface "inside," the outside interface "outside," and any perimeter interface "intf<i>n</i>" where <i>n</i> is 2 through 5.</p>
<i>security_level</i>	<p>Either <b>0</b> for the outside network or <b>100</b> for the inside network. Perimeter interfaces can use any number between <b>1</b> and <b>99</b>. By default, PIX Firewall sets the security level for the inside interface to <b>security100</b> and the outside interface to <b>security0</b>. The first perimeter interface is initially set to <b>security10</b>, the second to <b>security15</b>, the third to <b>security20</b>, and the fourth perimeter interface to <b>security25</b> (a total of 6 interfaces are permitted, with a total of 4 perimeter interfaces permitted).</p> <p>For access from a higher security to a lower security level, <b>nat</b> and <b>global</b> commands or <b>static</b> commands must be present. For access from a lower security level to a higher security level, <b>static</b> and <b>access-list</b> commands must be present.</p> <p>Interfaces with the same security level cannot communicate with each other. We recommend that every interface have a unique security level.</p>

## Usage Guidelines

The **nameif** command lets you assign a name to an interface. You can use this command to assign interface names if you have more than two network interface circuit boards in your PIX Firewall. The first two interfaces have the default names **inside** and **outside**. The **inside** interface has a default security level of 100, the **outside** interface has a default security level of 0.

## Usage Notes

1. If you change the *hardware\_id* of the outside interface; for example, from ethernet0 to ethernet1, PIX Firewall changes every reference to the outside interface in your configuration to inside, which can cause problems with **route**, **ip**, and other command statements that affect the flow of traffic through the PIX Firewall.
2. After changing a **nameif** command, use the **clear xlate** command.

3. The inside interface cannot be renamed or given a different security level. The outside interface can be renamed, but not given a different security level.
4. An interface is always “external” with respect to another interface that has a higher security level.
5. Up to 6 Ethernet interfaces are supported; up to 2 FDDI interfaces are supported. Refer to the *Installation Guide for the Cisco Secure PIX Firewall Version 5.2* for additional caveats.

See also: **interface**.

### Examples

The following example shows use of the **nameif** command:

```
nameif ethernet2 perimeter1 sec50
nameif ethernet3 perimeter2 sec20
```

# nat

Associate a network with a pool of global IP addresses. (Configuration mode.)

```
nat [(if_name)] nat_id local_ip [netmask [max_conns [em_limit]]] [norandomseq]
```

```
nat [(if_name)] 0 access-list acl_name
```

```
nat [(if_name)] 0 local_ip [netmask [max_conns [em_limit]]] [norandomseq]
```

```
no nat [[(if_name)] nat_id local_ip [netmask [max_conns [em_limit]]]] [norandomseq]
```

```
no nat [(if_name)] 0 access-list acl_name
```

```
show nat
```

## Syntax Description

<i>if_name</i>	The internal network interface name.  If the interface is to be associated with an access list, then the <i>if_name</i> is the higher security level interface name.
<i>nat_id</i>	All <b>nat</b> command statements with the same <i>nat_id</i> are in the same <b>nat</b> group. Use the <i>nat_id</i> in the <b>global</b> command statement; for example:  <pre>nat (perimeter) 1 0 0 global (outside) 1 209.165.201.1-209.165.201.30 netmask 255.255.255.224</pre> This example associates the <b>nat</b> command with the <b>global</b> command via the <i>nat_id</i> . The <i>nat_id</i> is an arbitrary positive number between 0 and two billion. This number can be the same as the ID used with the <b>outbound</b> and <b>apply</b> commands.  Specify <b>0</b> with IP addresses and netmasks to identify internal networks that desire only outbound identity address translation. Specify <b>0</b> with the <b>access-list</b> option to specify traffic that should be exempted from NAT.
<b>access-list</b>	Associate an <b>access-list</b> command statements to the <b>nat 0</b> command.
<i>local_ip</i>	Internal network IP address to be translated. You can use <b>0.0.0.0</b> to allow all hosts to start outbound connections. The <b>0.0.0.0</b> <i>local_ip</i> can be abbreviated as <b>0</b> .
<i>netmask</i>	Network mask for <i>local_ip</i> . You can use <b>0.0.0.0</b> to allow all outbound connections to translate with IP addresses from the global pool.
<i>max_conns</i>	The maximum TCP connections permitted from the interface you specify.
<i>em_limit</i>	The embryonic connection limit. The default is 0, which means unlimited connections. Set it lower for slower systems, higher for faster systems.
<b>norandomseq</b>	Do not randomize the TCP packet's sequence number. Only use this option if another inline firewall is also randomizing sequence numbers and the result is scrambling the data. Use of this option opens a security hole in the PIX Firewall.

### Usage Guidelines

The **nat** command lets you enable or disable address translation for one or more internal addresses. Address translation means that when a host starts an outbound connection, the IP addresses in the internal network are translated into global addresses. Network Address Translation (NAT) lets your network have any IP addressing scheme and the firewall protects these addresses from visibility on the external network.

The **nat (if\_name) 0 access-list acl\_name** command lets you exempt traffic that is matched by the **access-list** command statements from the NAT services. Adaptive Security remains in effect with the **nat 0 access-list** command. The extent to which the inside hosts are accessible from the outside depends on the **access-list** command statements that permit inbound access. The *if\_name* is the higher security level interface name. The *acl\_name* is the name you use to identify the **access-list** command statement.



#### Note

The new **access-list** option changes the behavior of the **nat 0** command. (Without the **access-list** option, the command is backward compatible with previous versions.) Previously, **nat 0** implemented the identity feature; this new version of the command disables NAT. Specifically, the new behavior disables proxy ARPing for the IP addresses in the **nat 0** command statement.



#### Note

The access list you specify with the **nat 0 access-list** command will not work with an **access-list** command statement that contains a port specification. The following sample command statements will not work:

```
access-list no-nat permit tcp host xx.xx.xx.xx host yy.yy.yy.yy
nat (inside) 0 access-list no-nat
```

After changing or removing a **nat** command statement, use the **clear xlate** command.

The connection limit lets you set the maximum number of outbound connections that can be started with the IP address criteria you specify. The embryonic connection limit lets you prevent a type of attack where processes are started without being completed. An embryonic connection is a connection that someone attempted but has not completed and has not yet seen data. Every connection is embryonic until it sets up.

You can use the **no nat** command to remove a **nat** command statement and you can use the **show nat** command to view **nat** command statements in the current configuration.

Table 5-8 helps you decide when to use the **nat** or **static** commands for access between the various interfaces in the PIX Firewall. For this table, assume that the security levels are 40 for dmz1 and 60 for dmz2.

**Table 5-8 Interface Access Commands by Interface**

From This Interface	To This Interface	Use This Command		From This Interface	To This Interface	Use This Command
inside	outside	<b>nat</b>		dmz2	outside	<b>nat</b>
inside	dmz1	<b>nat</b>		dmz2	dmz1	<b>nat</b>
inside	dmz2	<b>nat</b>		dmz2	inside	<b>static</b>
dmz1	outside	<b>nat</b>		outside	dmz1	<b>static</b>

**Table 5-8** Interface Access Commands by Interface (continued)

From This Interface	To This Interface	Use This Command		From This Interface	To This Interface	Use This Command
dmz1	dmz2	<b>static</b>		outside	dmz2	<b>static</b>
dmz1	inside	<b>static</b>		outside	inside	<b>static</b>

The rule of thumb is that for access from a higher security level interface to a lower security level interface, use the **nat** command. From lower security level interface to a higher security level interface, use the **static** command.

### Usage Notes

1. You can enable identity address translation with the **nat 0** command. Use this command when you have IP addresses that are the same as those used on more than one interface. Adaptive Security remains in effect with the **nat 0** command. The extent to which the inside hosts are accessible from the outside depends on the **access-list** command statements that permit inbound access.

Addresses on each interface must be on a different subnet. See Appendix D, “Subnet Masking and Addressing,” for more information on subnetting.

The **nat 0 10.2.3.0** command means let those IP addresses in the 10.2.3.0 net appear on the outside without translation. All other hosts are translated depending on how their **nat** command statements appear in the configuration.

2. The **nat 1 0 0** command means that all outbound connections can pass through the PIX Firewall with address translation. If you use the **nat (inside) 1 0 0** command, users can start connections on any interface with a lower security level, on the both perimeter interfaces and the outside interface. With NAT in effect, you must also use the **global** command statement to provide a pool of addresses through which translated connections pass. In effect, you use the **nat** command statement to specify from which interface connections can originate and you use the **global** command statement to determine at which interface connections can occur. The NAT ID must be the same on the **nat** and **global** command statements.
3. The **nat 1 10.2.3.0** command means that only outbound connections originating from the inside host 10.2.3.0 can pass through the firewall to go to their destinations with address translation.

See also: **global, outbound, apply.**

### Examples

The following example specifies with **nat** command statements that all the hosts on the 10.0.0.0 and 3.3.3.0 inside networks can start outbound connections. The **global** command statements create a pool of global addresses:

```
nat (inside) 1 10.0.0.0 255.0.0.0
global (outside) 1 209.165.201.25-209.165.201.27 netmask 255.255.255.224
global (outside) 1 209.165.201.30
```

```
nat (inside) 3 10.3.3.0 255.255.255.0
global (outside) 3 209.165.201.10-209.165.201.25 netmask 255.255.255.224
```

When using the **nat 0** command, if you want the addresses to be visible from the outside network, use **static** and **access-list** command statements:

```
nat (inside) 0 209.165.201.0 255.255.255.224
static (inside, outside) 209.165.201.0 209.165.201.0 netmask 255.255.255.224
access-list acl_out permit host 10.0.0.1 209.165.201.0 255.255.255.224 eq ftp
access-group acl_out in interface outside

nat (inside) 0 209.165.202.128 255.255.255.224
static (inside, outside) 209.165.202.128 209.165.202.128 netmask 255.255.255.224
access-list acl_out permit tcp host 10.0.0.1 209.165.202.128 255.255.255.224 eq ftp
access-group acl_out in interface outside
...
```

The following example shows use of the **nat 0 access-list** command to permit internal host 10.1.1.15, accessible through inside interface, “inside,” to bypass NAT when connecting to outside host 10.2.1.3.

```
access-list no-nat permit ip host 10.1.1.15 host 10.2.1.3
nat (inside) 0 access-list no-nat
```

The following will disable all NAT on a PIX Firewall with three interfaces:

```
access-list all-ip-packet permit ip 0 0 0 0
nat (dmz) 0 access-list all-ip-packet
nat (inside) 0 access-list all-ip-packet
```

# outbound/apply

Create an access list for controlling Internet use. (Configuration mode.)

**outbound** *list\_ID* **permit** | **deny** *ip\_address* [*netmask* [*port*[-*port*]]] [*protocol*]

**outbound** *list\_ID* **except** *ip\_address* [*netmask* [*port*[-*port*]]] [*protocol*]

**clear** **outbound**

**no** **outbound** [*list\_ID* **permit** | **deny** *ip\_address* [*netmask* [*port*[-*port*]]] [*protocol*]]

**no** **outbound** [*list\_ID* **except** *ip\_address* [*netmask* [*port*[-*port*]]] [*protocol*]]

**show** **outbound**

**apply** [(*if\_name*)] *list\_ID* **outgoing\_src** | **outgoing\_dest**

**clear** **apply**

**no** **apply** [(*if\_name*)] *list\_ID* **outgoing\_src** | **outgoing\_dest**

**show** **apply** [(*if\_name*)] [*list\_ID* **outgoing\_src** | **outgoing\_dest**]

## Syntax Description

<i>list_ID</i>	A tag number for the access list. The access list number you use must be the same for the <b>apply</b> and <b>outbound</b> commands. This value must be a positive number from 1 to 1599. This number can be the same as what you use with the <b>nat</b> and <b>global</b> commands. This number is just an arbitrary number that groups <b>outbound</b> command statements to an <b>apply</b> command statement.
<b>permit</b>	Allow the access list to access the specified IP address and port.
<b>deny</b>	Deny the access list access to the specified IP address and port.
<b>except</b>	<p>Create an exception to a previous <b>outbound</b> command. An <b>except</b> command statement applies to <b>permit</b> or <b>deny</b> command statements only with the same access list ID.</p> <p>When used with <b>apply outgoing_src</b>, the IP address of an <b>except</b> command statement applies to the destination address.</p> <p>When used with <b>apply outgoing_dest</b>, the IP address of an <b>except</b> command statement applies to the source address.</p> <p>See “Outbound List Rules” for more information.</p>
<i>ip_address</i>	The IP address for this access list entry. Do not specify a range of addresses. The 0.0.0.0 <i>ip_address</i> can be abbreviated as 0.
<i>netmask</i>	The network mask for comparing with the IP address; 255.255.255.0 causes the access list to apply to an entire Class C address. 0.0.0.0 indicates all access. The 0.0.0.0 <i>netmask</i> can be abbreviated as 0.

<i>port</i>	A port or range of ports that the access list is permitted or denied access to. See the “Ports” section in Chapter 1, “Introduction” for a list of valid port literal names.
<i>protocol</i>	Limit outbound access to <b>udp</b> , <b>tcp</b> , or <b>icmp</b> protocols. If a protocol is not specified, the default is <b>tcp</b> .
<i>if_name</i>	The network interface originating the connection.
<b>outgoing_src</b>	Deny or permit an internal IP address the ability to start outbound connections using the service(s) specified in the <b>outbound</b> command.
<b>outgoing_dest</b>	Deny or permit access to an external IP address using the service(s) specified in the <b>outbound</b> command.

### Usage Guidelines

The **outbound** command creates an access list that lets you specify the following:

- Whether inside users can create outbound connections
- Whether inside users can access specific outside servers
- What services inside users can use for outbound connections and for accessing outside servers
- Whether outbound connections can execute Java applets on the inside network

Outbound lists are filters on outgoing packets from the PIX Firewall. The filter can be based on the source IP address, the destination IP address, and the destination port/protocol as specified by the rules. The use of an **outbound** command requires use of the **apply** command. The **apply** command lets you specify whether the access control list applies to inside users’ ability to start outbound connections with **apply** command’s **outgoing\_src** option, or whether the access list applies to inside users’ ability to access servers on the outside network with the **apply** command’s **outgoing\_dest** option.



#### Note

The **outbound** command has been superseded by the **access-list** command. We recommend that you migrate your **outbound** command statements to **access-list** command statements to maintain future compatibility.



#### Note

The **java** option has been replaced by the **filter java** command.

After adding, removing, or changing **outbound** command statements, use the **clear xlate** command.

Use the **no outbound** command to remove a single **outbound** command statement from the configuration. Use the **clear outbound** command to remove all **outbound** command statements from the configuration. The **show outbound** command displays the **outbound** command statements in the configuration.

Use the **no apply** command to remove a single **apply** command statement from the configuration. Use the **clear apply** command statement to remove all the **apply** command statements from the configuration. The **show apply** command displays the **apply** command statements in the configuration.

### Outbound List Rules

Rules, written as **outbound** *list\_ID*... command statements are global to the PIX Firewall, they are activated by **apply** *list\_ID outgoing\_src | outgoing\_dest* command statements. When applied to *outgoing\_src*, the source IP address, the destination port, and protocol are filtered. When applied to *outgoing\_dest*, the destination IP address, port, and protocol are filtered.

The *outgoing\_src* and *outgoing\_dest* outbound lists are filtered independently. If any one of the filters contain **deny**, the outbound packet is denied. When multiple rules are used to filter the same packet, the best matched rule takes effect. The best match is based on the IP address mask and the port range check. More strict IP address masks and smaller port ranges are considered a better match. If there is a tie, a **permit** overrides a **deny**.

Rules are grouped by a *list\_ID*. Within each *list\_ID*, **except** rules (that is, **outbound n except** ...) can be set. The **except** option reverses the best matched rule of **deny** or **permit**. In addition, PIX Firewall filters the specified IP address and mask in the rule for the destination IP address of the outbound packet if the list is applied to the *outbound\_src*. Alternatively, PIX Firewall filters the source IP address if the list is applied to the *outgoing\_dest*. Furthermore, the **except** rules only apply to rules with the same *list\_ID*. A single **except** rule within a *list\_ID* without another **permit** or **deny** rule has no effect. If multiple **except** rules are set, the best match is checked for which **except** to apply.

The **outbound** command rules are now sorted by the best match checking. Use the **show outbound** command to see how the best match is judged by the PIX Firewall.

### Usage Notes

1. If **outbound** commands are not specified, the default behavior is to permit all outbound traffic and services from inside hosts.
2. After adding, changing, or removing an **outbound** and **apply** command statement group, use the **clear xlate** command to make the IP addresses available in the translation table.
3. The **outbound** commands are processed linearly within a *list\_ID*. In addition, *list\_IDs* are processed sequentially in descending order. For example, the first command statement you specify in an **outbound** list is processed first, then the next **outbound** command statement in that list, and so on. Similarly, *list\_ID* 10 is processed before *list\_ID* 20, and so on.
4. When using **outbound** commands, it is often helpful to deny or permit access to the many before you deny or permit access to the specific. Start with an interface-wide specification such as the following that denies all hosts from starting connections:

```
outbound 1 deny 0 0 0
apply (inside) 1 outgoing_src
```

Then add command statements that permit or deny hosts access to specific ports, for example:

```
outbound 1 deny 0 0 0
outbound 1 permit 10.1.1.1 255.255.255.255 23 tcp
outbound 1 permit 10.1.1.1 255.255.255.255 80 tcp
apply (inside) 1 outgoing_src
```

If you used the **except** option, you could state this same example as follows:

```
outbound 1 deny 0 0 0
outbound 1 except 209.165.201.11 255.255.255.255 23 tcp
outbound 1 except 209.165.201.11 255.255.255.255 80 tcp
apply (inside) 1 outgoing_src
```

In the preceding **outbound except** command statement, IP address 209.165.201.11 is the destination IP address, not the source address. This means that everyone is denied outbound access, except those users going to 209.165.201.11 via Telnet (port 23) or HTTP (port 80).

5. If you permit access to port 80 (**http**), this also permits Java applets to be downloaded. You must have a specific **deny** command statement to block Java applets.
6. The maximum number of **outbound** list entries in a configuration is 1599.
7. Outbound lists have no effect on **access-list** command statement groups.
8. The use of the **access-group** command statement overrides the **conduit** and **outbound** command statements for the specified interface name.

### Examples

The first **outbound** group sets inside hosts so that they can only see and Telnet to perimeter hosts, and do DNS lookups. In this example, the perimeter network address is 209.165.201.0 and the network mask is 255.255.255.224:

```
outbound 9 deny 0.0.0.0 0.0.0.0 0 0
outbound 9 except 209.165.201.0 255.255.255.224 23 tcp
outbound 9 except 0.0.0.0 0.0.0.0 53 udp
```

The next **outbound** group in this same example lets hosts 10.1.1.11 and 10.1.1.12 go anywhere:

```
outbound 11 deny 0.0.0.0 0.0.0.0 0 0
outbound 11 permit 10.1.1.11 255.255.255.255 0 0
outbound 11 permit 10.1.1.12 255.255.255.255 0 0
outbound 11 permit 0.0.0.0 0.0.0.0 21 tcp
outbound 11 permit 10.3.3.3 255.255.255.255 143 tcp
```

This last **outbound** group in this same example lets hosts on the perimeter only access TCP ports 389 and 30303 and UDP port 53 (DNS). Finally, the **apply** command statements set the **outbound** groups so that the permit and deny rules affect access to all external addresses.

```
outbound 13 deny 0.0.0.0 0.0.0.0 0 0
outbound 13 permit 0.0.0.0 0.0.0.0 389 tcp
outbound 13 permit 0.0.0.0 0.0.0.0 30303 tcp
outbound 13 permit 0.0.0.0 0.0.0.0 53 udp
```

```
apply (inside) 9 outgoing_src
apply (inside) 11 outgoing_src
apply (perim) 13 outgoing_src
```

### Controlling Outbound Connections

The following example prevents all inside hosts from starting outbound connections:

```
outbound 1 deny 0 0 0
apply (inside) 1 outgoing_src
```

The **0 0 0** at the end of the command means all IP addresses (**0** is the same as **0.0.0.0**), with a 0.0.0.0 subnet mask and for all services (port value is zero).

Conversely, the following example permits all inside hosts to start connections to the outside (this is the default if an access list is not created):

```
outbound 1 permit 0 0 0
apply (inside) 1 outgoing_src
```

### Controlling Inside Hosts' Access to Outbound Services

The following example prevents inside host 192.168.1.49 from accessing the World Wide Web (port 80):

```
outbound 11 deny 192.168.1.49 255.255.255.255 80 tcp
apply (inside) 11 outgoing_src
```

### Controlling Inside Hosts' Access to Outside Servers

If your employees are spending too much time examining GIF images on a particular site with two web servers, you can use the following example to restrict this access:

```
outbound 12 deny 192.168.146.201 255.255.255.255 80 tcp
outbound 12 deny 192.168.146.202 255.255.255.255 80 tcp
apply (inside) 12 outgoing_dest
```

### Using except Command Statements

An **except** command statement only provides exception to items with the same *list\_ID*. Consider the following example:

```
outbound 9 deny 0.0.0.0 0.0.0.0 0 0
outbound 9 except 10.100.0.0 255.255.0.0 23 tcp
outbound 9 except 0.0.0.0 0.0.0.0 53 udp
outbound 11 deny 0.0.0.0 0.0.0.0 0 0
outbound 11 permit 10.1.1.11 255.255.255.255 0 0
outbound 11 permit 10.1.1.12 255.255.255.255 0 0
outbound 11 permit 0.0.0.0 0.0.0.0 21 tcp
outbound 11 permit 10.3.3.3 255.255.255.255 143 tcp
outbound 13 deny 0.0.0.0 0.0.0.0 0 0
outbound 13 permit 0.0.0.0 0.0.0.0 389 tcp
outbound 13 permit 0.0.0.0 0.0.0.0 30303 tcp
outbound 13 permit 0.0.0.0 0.0.0.0 53 udp
```

In the preceding examples, the following two command statements work against other command statements in list 9 but not in lists 11 and 13:

```
outbound 9 except 10.100.0.0 255.255.0.0 23 tcp
outbound 9 except 0.0.0.0 0.0.0.0 53 udp
```

In the following example, the set of **deny**, **permit**, and **except** option command statements denies everybody from connecting to external hosts except for DNS queries and Telnet connections to hosts on 10.100.0.0. The host with IP address 10.1.1.11 is permitted outbound access, and has access to everywhere *except* to 10.100.0.0 via Telnet and anywhere to use DNS:

```
outbound 1 deny 0.0.0.0 0.0.0.0 0 tcp
outbound 1 permit 10.1.1.11 255.255.255.255 0 tcp
outbound 1 except 10.100.0.0 255.255.0.0 23 tcp
outbound 1 except 0.0.0.0 0.0.0.0 53 udp
apply (inside) outgoing_src
```

# pager

Enable or disable screen paging. (Privileged mode.)

**pager** [**lines** *number*]

**clear pager**

**no pager**

**show pager**

## Syntax Definition

*number* The number of lines before the More prompt appears. The minimum is **1**.  
Use **0** to disable paging.

## Usage Guidelines

The **pager lines** command lets you specify the number of lines in a page before the More prompt appears. The **pager** command enables display paging, and **no pager** disables paging and lets output display completely without interruption. If you set **pager lines** to some value and want to revert back to the default, enter the **pager** command without options. The **clear pager** command resets the number of lines in a page to 24.

Use **pager 0** to disable paging.

The **show pager** command displays **pager** status.

When paging is enabled, the following prompt appears:

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

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

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

To return to the command line, press the **q** key.

## Examples

The following example shows use of the **pager** command:

```
pixfirewall# pager lines 2
pixfirewall# ping inside 10.0.0.42
    10.0.0.42 NO response received -- 1010ms
    10.0.0.42 NO response received -- 1000ms
<--- More --->
```

# passwd

Set password for Telnet and PIX Firewall Manager access to the firewall console. (Privileged mode.)

**passwd** *password* [**encrypted**]

**clear passwd**

**show passwd**

## Syntax Description

*password* A case-sensitive password of up to 16 alphanumeric and special characters. Any character can be used in the password except a question mark and a space.

**encrypted** Specifies that the password you entered is already encrypted. The *password* you specify with the **encrypted** option must be 16 characters in length.

## Usage Guidelines

The **passwd** command sets a password for Telnet and PIX Firewall Manager access to the firewall console. An empty password is also changed into an encrypted string. However, any use of a **write** command displays or writes the passwords in encrypted form. Once passwords are encrypted, they are not reversible back to plain text. The **clear passwd** command resets the password to “cisco.”



### Note

---

Write down the new password and store it in a manner consistent with your site’s security policy. Once you change this password, you cannot view it again.

---

See also: **enable password**.

## Examples

The following example shows use of the **passwd** command:

```
passwd watag00slam
show passwd
passwd jMorNbK0514fadBh encrypted
```

# perfmon

View performance information. (Privileged mode.)

**perfmon interval** *seconds*

**perfmon quiet** | **verbose**

**show perfmon**

## Syntax Description

**interval** *seconds* Specify the number of seconds between when the performance displays appear on the console. The default is 120 seconds.

**quiet** Disable performance monitor displays.

**verbose** Enable displaying performance monitor information at the PIX Firewall console.

## Usage Guidelines

The **perfmon** command lets you monitor the PIX Firewall unit's performance. Use the **show perfmon** command to view the information immediately. Use the **perfmon verbose** command to display the information every two minutes continuously. Use the **perfmon interval** *seconds* command with the **perfmon verbose** command to display the information continuously every number of seconds you specify.



### Note

---

The **show perfmon** command does not display in a Telnet console session.

---

Use the **perfmon quiet** command to disable the display.

An example of the performance information follows:

PERFMON STATS:	Current	Average
Xlates	33/s	20/s
Connections	110/s	10/s
TCP Conns	50/s	42/s
WebSns Req	4/s	2/s
TCP Fixup	20/s	15/s
HTTP Fixup	5/s	5/s
FTP Fixup	7/s	4/s
AAA Authen	10/s	5/s

AAA Author	9/s	5/s
AAA Account	3/s	3/s

This information lists the number of translations, connections, Websense requests, address translations (called “fixups”), and AAA transactions that occur each second.

### Examples

The following commands display the performance monitor statistics every 30 seconds on the PIX Firewall console:

```
perfmon interval 30
perfmon verbose
```

# ping

Determine if other IP addresses are visible from the PIX Firewall. (Privileged mode.)

```
ping [if_name] ip_address
```

## Syntax Description

*if\_name* The internal or external network interface name. The address of the specified interface is used as the source address of the ping.

*ip\_address* The IP address of a host on the inside or outside networks.

## Usage Guidelines

The **ping** command determines if the PIX Firewall has connectivity or if a host is available on the network. The command output shows if the response was received; that is, that a host is participating on the network. If a host is not responding, **ping** displays “NO response received.” Use the **show interface** command to ensure that the PIX Firewall is connected to the network and is passing traffic.

If you want internal hosts to be able to ping external hosts, you must create an ICMP **access-list** command statement for echo reply; for example, to give ping access to all hosts, use the **access-list *acl\_grp* permit icmp any any** command and bind the **access-list** command statement to the interface you want to test using an **access-group** command statement. Refer to “Step 8—Permit Ping Access” in Chapter 2, “Configuring the PIX Firewall,” for more information on these commands.

If you are pinging through PIX Firewall between hosts or routers, but the pings are not successful, use the **debug icmp trace** command to monitor the success of the ping. If pings are both inbound and outbound, they are successful.

The PIX Firewall **ping** command no longer requires an interface name. If an interface name is not specified, PIX Firewall checks the routing table to find the address you specify. You can specify an interface name to indicate through which interface the ICMP echo requests are sent.

An example of the new usage is as follows:

```
ping 10.0.0.1
10.0.0.1 response received -- 10ms
10.0.0.1 response received -- 10ms
10.0.0.1 response received -- 0ms
```

Or you can still enter the command as before:

```
ping outside 10.0.0.1
10.0.0.1 response received -- 10ms
10.0.0.1 response received -- 10ms
10.0.0.1 response received -- 0ms
```

## Examples

The **ping** command makes three attempts to reach an IP address:

```
ping 192.168.42.54
192.168.42.54 response received -- 0ms
192.168.42.54 response received -- 0ms
192.168.42.54 response received -- 0ms
```

# quit

Exit configuration or privileged mode. (All modes.)

## **quit**

### **Usage Guidelines**

Use the **quit** command to exit configuration or privileged mode.

### **Examples**

The following example shows use of the **quit** command:

```
pixfirewall(config)# quit  
pixfirewall# quit  
pixfirewall>
```

# reload

Reboot and reload the configuration. (Privileged mode.)

## reload

### Usage Guidelines

The **reload** command reboots the PIX Firewall and reloads the configuration from a bootable floppy disk or, if a diskette is not present, from Flash memory.



#### Note

---

You are prompted for confirmation before starting with “Proceed with reload?”. Any response other than **n** causes the reboot to occur.

---



#### Note

---

Configuration changes not written to Flash memory are lost after reload. Before rebooting, store the current configuration in Flash memory with the **write memory** command.

---

### Examples

The following example shows use of the **reload** command:

```
reload
Proceed with reload? [confirm] y

Rebooting...

PIX Bios V2.7
...
```

# rip

Change RIP settings. (Configuration mode.)

```
rip if_name default | passive [version [1 | 2]] [authentication [text | md5 key (key_id)]]
no rip if_name default | passive [version [1 | 2]] [authentication [text | md5 key (key_id)]]
show rip [if_name]
clear rip
```

## Syntax Description

<i>if_name</i>	The internal or external network interface name.
<b>default</b>	Broadcast a default route on the interface.
<b>passive</b>	Enable passive RIP on the interface. The PIX Firewall listens for RIP routing broadcasts and uses that information to populate its routing tables.
<b>version</b>	RIP version. Use <b>version 2</b> for RIP update encryption. Use <b>version 1</b> to provide backward compatibility with the older version.
<b>authentication</b>	Enable RIP version 2 authentication.
<b>text</b>	Send RIP updates as clear text (not recommended).
<b>md5</b>	Send RIP updates using MD5 encryption.
<i>key</i>	Key to encrypt RIP updates. This value must be the same on the routers and any other device <i>that provides RIP version 2 updates</i> . The <i>key</i> is a text string of up to 16 characters in length.
<i>key_id</i>	Key identification value. The <i>key_id</i> can be a number from 1 to 255. Use the same <i>key_id</i> in use on the routers and any other device <i>that provides RIP version 2 updates</i> .

## Usage Guidelines

The **rip** command enables IP routing table updates from received RIP (Routing Information Protocol) broadcasts. Use the **show rip** command to display the current RIP settings. Use the **no rip** command to disable the PIX Firewall IP routing table updates. The default is to enable IP routing table updates. If you specify RIP version 2, you can encrypt RIP updates using MD5 encryption.

The **clear rip** command removes all the **rip** commands from the configuration.



### Note

Ensure that the *key* and *key\_id* values are the same as in use on any other device in your network that makes RIP version 2 updates.

**Note**


---

The PIX Firewall cannot pass RIP updates between interfaces.

---

**Examples**

The following is sample output from the version 1 **show rip** and **rip inside default** commands:

```
show rip
rip outside passive
no rip outside default
rip inside passive
no rip inside default
```

```
rip inside default
show rip
rip outside passive
no rip outside default
rip inside passive
rip inside default
```

The next example combines version 1 and version 2 commands and shows listing the information with the **show rip** command after entering the rip commands that:

- Enable version 2 passive RIP using MD5 authentication on the outside interface to encrypt the key used by the PIX Firewall and other RIP peers, such as routers.
- Enable version 1 passive RIP listening on the inside interface of the PIX Firewall.
- Enable version 2 passive RIP listening on the dmz interface of the PIX Firewall.

```
rip outside passive version 2 authentication md5 thisisakey 2
rip outside default version 2 authentication md5 thisisakey 2
rip inside passive
rip dmz passive version 2
```

```
show rip
rip outside passive version 2 authentication md5 thisisakey 2
rip outside default version 2 authentication md5 thisisakey 2
rip inside passive version 1
rip dmz passive version 2
```

The next example shows how use of the **clear rip** command clears all the previous rip commands from the current configuration:

```
clear rip
show rip
```

This example shows use of the version 2 feature that passes the encryption key in text form:

```
rip out default version 2 authentication text thisisakey 3
show rip
rip outside default version 2 authentication text thisisakey 3
```

# route

Enter a static or default route for the specified interface. (Configuration mode.)

```
route if_name ip_address netmask gateway_ip [metric]
```

```
clear route [if_name ip_address [netmask gateway_ip]]
```

```
no route [if_name ip_address [netmask gateway_ip]]
```

```
show route
```

## Syntax Description

<i>if_name</i>	The internal or external network interface name.
<i>ip_address</i>	The internal or external network IP address. Use <b>0.0.0.0</b> to specify a default route. The <b>0.0.0.0</b> IP address can be abbreviated as <b>0</b> .
<i>netmask</i>	Specify a network mask to apply to <i>ip_address</i> . Use <b>0.0.0.0</b> to specify a default route. The <b>0.0.0.0</b> <i>netmask</i> can be abbreviated as <b>0</b> .
<i>gateway_ip</i>	Specify the IP address of the gateway router (the next hop address for this route).
<i>metric</i>	Specify the number of hops to <i>gateway_ip</i> . If you are not sure, enter <b>1</b> . Your network administrator can supply this information or you can use a <b>traceroute</b> command to obtain the number of hops. The default is <b>1</b> if a metric is not specified.

## Usage Guidelines

Use the **route** command to enter a default or static route for an interface. To enter a default route, set *ip\_address* and *netmask* to **0.0.0.0**, or the shortened form of **0**. All routes entered using the **route** command are stored in the configuration when it is saved. The **clear route** command removes **route** command statements from the configuration that do not contain the CONNECT keyword.

Create static routes to access networks connected outside a router on any interface. The effect of a static route is like stating “to send a packet to the specified network, give it to this router.” For example, PIX Firewall sends all packets destined to the 192.168.42.0 network through the 192.168.1.5 router with this static **route** command statement:

```
route dmz 192.168.42.0 255.255.255.0 192.168.1.5 1
```

The routing table automatically specifies the IP address of a PIX Firewall interface in the **route** command. Once you enter the IP address for each interface, PIX Firewall creates a **route** statement entry that is not deleted when you use the **clear route** command.

If the **route** command statement uses the IP address from one of the PIX Firewall unit’s interfaces as the gateway IP address, PIX Firewall will ARP for the destination IP address in the packet instead of ARPing for the gateway IP address.

The following steps show how PIX Firewall handles routing:

- 
- Step 1** PIX Firewall receives a packet from the inside interface destined to IP address X.
  - Step 2** Because a default route is set to itself, PIX Firewall sends out an ARP for address X.
  - Step 3** Any Cisco router on the outside interface LAN which has a route to address X (Cisco IOS software has proxy ARP enabled by default) replies back to the PIX Firewall with its own MAC address as the next hop.
  - Step 4** PIX Firewall sends the packet to router (just like a default gateway).
  - Step 5** PIX Firewall adds the entry to its ARP cache for IP address X with the MAC address being that of the router.

- The CONNECT route entry is supported. (This identifier appears when you use the **show route** command.) The CONNECT identifier is assigned to an interface's local network and the interface IP address, which is in the IP local subnet. PIX Firewall will ARP for the destination address. The CONNECT identifier cannot be removed, but changes when you change the IP address on the interface.
- If you enter duplicate routes with different metrics for the same gateway, PIX Firewall changes the metric for that route and updates the metric for the route.

For example, the following command statement is in a configuration:

```
route inside 10.0.0.0 255.0.0.0 10.0.0.2 2 OTHER
```

If you enter the following statement:

```
route inside 10.0.0.0 255.0.0.0 10.0.0.2 3
```

PIX Firewall converts the command statement to the following:

```
route inside 10.0.0.0 255.0.0.0 10.0.0.2 3 OTHER
```

---

### Examples

Specify one default **route** command statement for the outside interface, which in this example, is for the router on the outside interface that has an IP address of 209.165.201.1:

```
route outside 0 0 209.165.201.1 1
```

For static routes, if two networks, 10.1.2.0 and 10.1.3.0 connect via a hub to the dmz1 interface router at 10.1.1.4, add these static **route** command statements to provide access to the networks:

```
route dmz1 10.1.2.0 255.0.0.0 10.1.1.4 1
route dmz1 10.1.3.0 255.0.0.0 10.1.1.4 1
```

# service

Reset inbound connections. (Configuration mode.)

**service resetinbound**

**show service**

## Syntax Description

**resetinbound** Reset inbound connections.

## Usage Guidelines

The **service** command works with all inbound TCP connections to statics whose access lists or uauth (user authorization) do not allow inbound. One use is for resetting IDENT connections. If an inbound TCP connection is attempted and denied, you can use the **service resetinbound** command to return an RST (reset flag in the TCP header) to the source. Without the option, the PIX Firewall drops the packet without returning an RST.

For use with IDENT, the PIX Firewall sends a TCP RST to the host connecting inbound and stops the incoming IDENT process so that email outbound can be transmitted without having to wait for IDENT to time out. In this case, the PIX Firewall sends a syslog message stating that the incoming connection was a denied connection. Without **service resetinbound**, the PIX Firewall drops packets that are denied and generates a syslog message stating that the SYN was a denied connection. However, outside hosts keep retransmitting the SYN until the IDENT times out.

When an IDENT connection is timing out, you will notice that connections slow down. Perform a trace to determine that IDENT is causing the delay and then invoke the **service** command.

The **service resetinbound** command provides a safer way to handle an IDENT connection through the PIX Firewall. Ranked in order of security from most secure to less secure are these methods for handling IDENT connections:

1. Use the **service resetinbound** command.
2. Use the **established** command with the **permitto tcp 113** options.
3. Enter **static** and **access-list** command statements to open TCP port 113.

When using the **aaa** command, if the first attempt at authorization fails and a second attempt causes a timeout, use the **service resetinbound** command to reset the client that failed the authorization so that it will not retransmit any connections. An example authorization timeout message in Telnet follows:

```
Unable to connect to remote host: Connection timed out
```

## Examples

The following example shows use of the **service resetinbound** command:

```
service resetinbound
show service
service resetinbound
```

# session

Access an embedded AccessPro router console. (Privileged mode.)



Note

---

The PIX 506 and PIX 515 do not support use of the **session** command.

---

**session enable**

**no session**

**show session**



Note

---

Only use this command if you have an AccessPro router installed in your PIX Firewall.

---

## Syntax Description

**enable** Enable the **session** command for communications with the AccessPro router.

## Usage Guidelines

The **session** command lets you specify Cisco IOS software commands on an AccessPro router console when the router is installed in your PIX Firewall. Use COM port 4 on the AccessPro router to communicate with the PIX Firewall.

Exit the router console session by entering tilde-dot (~.). Press the tilde key and when you hear a bell sound from your terminal, press the dot key.

While a router console session is occurring, the PIX Firewall disables failover because they both require the same interrupts.

## Examples

This example enables an AccessPro session, starts the session, and then disables it:

```

session enable
Session has been enabled.
session

Warning: FAILOVER has been disabled!!!
Attempting session with embedded router, use ~. to quit!

acpro> ~.

no session
Session has been disabled
session
Session is not enabled

```

# show

View command information. (Differs by mode.)

## **show ?**

### Usage Guidelines

The **show** command without arguments or the **show ?** command lets you view the names of the **show** commands and their descriptions. Explanations for each **show** command are provided on the respective command page for the command itself where appropriate; for example, **show arp** is described on the **arp** command page.



### Note

---

The **show** commands that do not have a command equivalent shown in this section are described on their respective command pages; for example, the **show interface** command is described on the **interface** command page.

---

If the **pager** command is enabled and when 24 lines display, the listing pauses, and the following prompt appears:

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

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

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

### Examples

The following is sample output for the **show ?** command:

```
show ?  
?          help ...
```

# show blocks/clear blocks

Show system buffer utilization. (Privileged mode.)

**clear blocks**

**show blocks**

## Usage Guidelines

The **show blocks** command lists preallocated system buffer utilization. In the **show blocks** listing, the SIZE column displays the block type. The MAX column is the maximum number of allocated blocks. The LOW column is the fewest blocks available since last reboot. The CNT column is the current number of available blocks. A zero in the LOW column indicates a previous event where memory exhausted. A zero in the CNT column means memory is exhausted now. Exhausted memory is not a problem as long as traffic is moving through the PIX Firewall. You can use the **show conn** command to see if traffic is moving. If traffic is not moving and the memory is exhausted, a problem may be indicated.

The **clear blocks** command keeps the maximum count to whatever number is allocated in the system and equates the low count to the current count.

You can also view the information from the **show blocks** command using SNMP. Refer to “Using the Firewall and Memory Pool MIBs” in Chapter 3, “Advanced Configurations,” for more information.

## Examples

The following is sample output for the **show blocks** command:

```
show blocks
  SIZE  MAX  LOW  CNT
    4   1600 1600 1600
   80    100  97   97
  256    80   79   79
 1550   788  402  404
65536    8    8    8
```

# show checksum

Display the configuration checksum. (Unprivileged mode.)

**show checksum**

## Usage Guidelines

The **show checksum** command displays four groups of hexadecimal numbers that act as a digital summary of the contents of the configuration. This same information stores with the configuration when you store it in Flash memory. By using the **show config** command and viewing the checksum at the end of the configuration listing and using the **show checksum** command, you can compare the numbers to see if the configuration has changed. The PIX Firewall tests the checksum to determine if a configuration has not been corrupted.

## Examples

The following is sample output for the **show checksum** command:

```
show checksum
Cryptochecksum: 1a2833c0 129ac70b 1a88df85 650dbb81
```

# show conn

Display all active connections. (Privileged mode.)

```
show conn [count] [foreign | local ip [-ip2] [netmask mask]] [protocol tcp | udp | protocol]
          [fport | lport port1 [-port2]] [state [up [,finin] [,finout] [,http_get] [,sip] [,smtp_data]
          [,smtp_banner] [,smtp_incomplete] [,nojava] [,data_in] [,data_out] [,sqlnet_fixup_data]
          [,conn_inbound] [,rpc] [,h323] [,dump]]
```

## Syntax Description

<b>count</b>	Display only the number of used connections. This feature is no longer supported and returns unreliable information.
<b>foreign   local ip [-ip2] netmask mask</b>	Display active connections by the foreign IP address or by local IP address. Qualify foreign or local active connections by network mask.
<b>protocol tcp   udp   protocol</b>	Display active connections by protocol type. <i>protocol</i> is a protocol specified by number. See “Protocols” in Chapter 1, “Introduction,” for a list of valid protocol literal names.

**fport | lport port1 [-port2]** Display foreign or local active connections by port. See “Ports” in Chapter 1, “Introduction,” for a list of valid port literal names.

**state** Display active connections by their current state: up (**up**), FIN inbound (**finin**), FIN outbound (**finout**), HTTP get (**http\_get**), SMTP mail data (**smtp\_data**), SIP connection (**sip**), SMTP mail banner (**smtp\_banner**), incomplete SMTP mail connection (**smtp\_incomplete**), an **outbound** command denying access to Java applets (**nojava**), inbound data (**data\_in**), outbound data (**data\_out**), SQL\*Net data fix up (**sqlnet\_fixup\_data**), inbound connection (**conn\_inbound**), RPC connection (**rpc**), H.323 connection (**h323**), dump clean up connection (**dump**).

### Usage Guidelines

The **show conn** command displays the number and information about the active TCP connections.

You can also view the connection count information from the **show conn** command using SNMP. Refer to “Using the Firewall and Memory Pool MIBs” in Chapter 3, “Advanced Configurations,” for more information.

### Examples

The following is sample output for the **show conn** command:

```
show conn
6 in use, 6 most used
TCP out 209.165.201.1:80 in 10.3.3.4:1404 idle 0:00:00 Bytes 11391
TCP out 209.165.201.1:80 in 10.3.3.4:1405 idle 0:00:00 Bytes 3709
TCP out 209.165.201.1:80 in 10.3.3.4:1406 idle 0:00:01 Bytes 2685
TCP out 209.165.201.1:80 in 10.3.3.4:1407 idle 0:00:01 Bytes 2683
TCP out 209.165.201.1:80 in 10.3.3.4:1403 idle 0:00:00 Bytes 15199
TCP out 209.165.201.1:80 in 10.3.3.4:1408 idle 0:00:00 Bytes 2688
UDP out 209.165.201.7:24 in 10.3.3.4:1402 idle 0:01:30
UDP out 209.165.201.7:23 in 10.3.3.4:1397 idle 0:01:30
UDP out 209.165.201.7:22 in 10.3.3.4:1395 idle 0:01:30
```

In this example, host 10.3.3.4 on the inside has accessed a web site at 209.165.201.1. The global address on the outside interface is 209.165.201.7.

## show history

Display previously entered lines. (Privileged mode.)

### show history

### Usage Guidelines

The **show history** command displays previously entered commands. You can examine commands individually with the up and down arrows or by entering **^p** to view previously entered lines or **^n** to view the next line.

**Examples**

The following is sample output for the **show history** command:

```
show history
  enable
  ...
```

## show interface

See the **interface** command page for a description of the **show interface** command.

## show memory

Show system memory utilization. (Privileged mode.)

```
show memory
```

**Usage Guidelines**

The **show memory** command displays a summary of the maximum physical memory and current free memory available to the PIX Firewall operating system. Memory in the PIX Firewall is allocated as needed.

You can also view the information from the **show memory** command using SNMP. Refer to “Using the Firewall and Memory Pool MIBs” in Chapter 3, “Advanced Configurations,” for more information.

**Examples**

The following is sample output for the **show memory** command:

```
show memory
nnnnnnnn bytes total, nnnnnnn bytes free
```

## show processes

Display processes. (Privileged mode.)

```
show processes
```

**Usage Guidelines**

The **show processes** command displays a listing of running processes. Processes are lightweight threads requiring only a few instructions. In the listing, PC is the program counter, SP is the stack pointer, STATE is the address of a thread queue, Runtime is the number of milliseconds that the thread has been running, SBASE is the stack base address, Stack is the current number of bytes used and the total size of the stack, and Process lists the thread’s function.

**Examples**

The following is sample output for the **show processes** command:

```
show processes
  PC      SP      STATE      Runtime      SBASE      Stack Process
Lsi 800125de 803603d0 80075ba0      0 8035f410 4004/4096 arp_timer
...
```

## show tech-support

View information to help a support analyst. (Privileged mode.)

**show tech-support**

### Usage Guidelines

The **show tech-support** command lists information technical support analysts need to help you diagnose PIX Firewall problems. This command combines the output from the **show** commands that provide the most information to a technical support analyst.

### Examples

The following is sample output for the **show tech-support** command:

```
show tech-support
PIX Version 5.2(n)nnn
Compiled on Fri 28-May-99 04:08 by pixbuild
PIX Bios V2.7

pixfirewall up 100 days 6 hours 17 mins
...
```

## show traffic/clear traffic

Shows interface transmit and receive activity. (Privileged mode.)

**clear traffic**

**show traffic**

### Usage Guidelines

The **show traffic** command lists the number of packets and bytes moving through each interface. The number of seconds is the duration the PIX Firewall has been online since the last reboot. The **clear traffic** command clears counters for the **show traffic** command output.

### Examples

The following is sample output for the **show traffic** command:

```
show traffic
outside:
  received (in 3786 secs):
    97 packets      6191 bytes
    42 pkts/sec    1 bytes/sec
  transmitted (in 3786 secs):
    99 packets      10590 bytes
    0 pkts/sec     2 bytes/sec ...
```

## show uauth

See the **uauth** command page for information on the **show uauth** command.

# show version

View the PIX Firewall operating information. (Unprivileged mode.)

## show version

### Usage Guidelines

The **show version** command lets you view the PIX Firewall unit's software version, operating time since last reboot, processor type, Flash memory type, interface boards, serial number (BIOS ID), and activation key value.



#### Note

The serial number listed with the **show version** command in version 5.2 and later is for the Flash memory BIOS. This number is different from the serial number on the chassis. When you get a software upgrade, you will need the serial number that appears in the **show version** command, not the chassis number.

In the following examples, the amount of Flash memory (2 MB or 16 MB) is identified by:

- Flash AT29C040A @ 0x300 for 2 MB of Flash
- Flash i28F640J5 @ 0x300 for 16 MB of Flash

### Examples

The following is sample output for the **show version** command.

#### show version

```
Cisco Secure PIX Firewall Version 5.2(1)
Compiled on Fri 01-Oct-99 13:56 by pixbuild

pix515 up 4 days 22 hours 10 mins 42 secs

Hardware:  PIX-515, 64 MB RAM, CPU Pentium 200 MHz
Flash i28F640J5 @ 0x300
BIOS Flash AT29C257 @ 0xffffd8000

0: ethernet0: address is 00aa.0000.0037, irq 11
1: ethernet1: address is 00aa.0000.0038, irq 10
2: ethernet2: address is 00a0.c92a.f029, irq 9
3: ethernet3: address is 00a0.c948.45f9, irq 7

Licensed Features:
Failover:      Enabled
VPN-DES:      Enabled
VPN-3DES:     Disabled
Maximum Interfaces: 6

Serial Number: 123 (0x7b)
Activation Key: 0xc5233151 0xb429f6d0 0xda93739a 0xe15cdf51
```

# show xlate

See the **xlate** command page for information on the **show xlate** command.

# snmp-server

Provide SNMP event information. (Configuration mode.)

**snmp-server community** *key*

**snmp-server contact** *text*

**snmp-server location** *text*

**snmp-server host** [*if\_name*] *ip\_addr*

**snmp-server enable traps**

**clear snmp-server** *command*

**no snmp-server** *command*

**show snmp-server**

## Syntax Description

- community** *key* Enter the password key value in use at the SNMP management station. The SNMP community string is a shared secret among the SNMP management station and the network nodes being managed. PIX Firewall uses the key to determine if the incoming SNMP request is valid. For example, you could designate a site with a community string and then configure the routers, firewall, and the management station with this same string. The PIX Firewall then honors SNMP requests using this string and does not respond to requests with an invalid community string. The *key* is a case-sensitive value up to 32 characters in length. Spaces are not permitted. The default, if this option is not used, is **public**.
- contact** *text* Supply your name or that of the PIX Firewall system administrator. The text is case-sensitive and can be up to 127 characters. Spaces are accepted, but multiple spaces are shortened to a single space.
- location** *text* Specify your PIX Firewall location. The text is case-sensitive and can be up to 127 characters. Spaces are accepted, but multiple spaces are shortened to a single space.
- host** Specify an IP address of the SNMP management station to which traps should be sent and/or from which the SNMP requests come. You can specify up to five SNMP management stations.
- Use with these parameters:
- *if\_name*—The interface name where the SNMP management station resides.
  - *ip\_addr*—The IP address of a host to which SNMP traps should be sent and/or from which the SNMP requests come.
- enable traps** Enable or disable sending SNMP trap notifications via syslog.

### Usage Guidelines

Use the **snmp-server** command to identify site, management station, community string, and user information.

In understanding SNMP use, the PIX Firewall is considered the SNMP agent or SNMP server. The management station is the system running the SNMP program that receives and processes the SNMP information that the PIX Firewall sends.

An SNMP object ID (OID) for PIX Firewall now displays in SNMP event traps sent from the PIX Firewall. OID 1.3.6.1.4.1.9.1.227 was assigned as the PIX Firewall system object ID.

The **clear snmp-server** and **no snmp-server** commands removes command statements. The **show snmp-server** command displays the information.

### MIB Support

You can browse the System and Interface groups of MIB-II. All SNMP values in the PIX Firewall are read only (RO). The PIX Firewall does not support browsing of the Cisco syslog MIB.

Browsing a MIB is different from sending traps. Browsing means doing an **snmpget** or **snmpwalk** of the MIB tree from the management station to determine values. Traps are different; they are unsolicited “comments” from the managed device to the management station for certain events, such as link up, link down, syslog event generated, and so on.

The Cisco Firewall MIB and Cisco Memory Pool MIB are now available. These MIBs provide the following PIX Firewall information via SNMP:

- Buffer usage from the **show block** command
- Connection count from the **show conn** command
- Failover status
- Memory usage from the **show memory** command

See “SNMP Traps” in Chapter 3, “Advanced Configurations” for more information on SNMP traps.

### Receiving SNMP Requests from an SNMP Management Station

To receive SNMP requests from a management station:

- 
- Step 1** Identify the management station with an **snmp-server host** command statement.
  - Step 2** Specify **snmp-server** command options for the **location**, **contact**, and **community**.
  - Step 3** Start the SNMP software on the management station and begin issuing SNMP requests to the PIX Firewall.
-

### Examples

The following example shows commands you would enter to start receiving SNMP requests from a management station:

```
snmp-server community wallawallabingbang
snmp-server location Building 42, Sector 54
snmp-server contact Sherlock Holmes
snmp-server host perimeter 10.1.2.42
```

The next example is sample output for the **show snmp-server** command:

```
show snmp
snmp-server host perimeter 10.1.2.42
snmp-server location Building 42, Sector 54
snmp-server contact Sherlock Holmes
snmp-server community wallawallabingbang
```

# ssh

Specify a host for PIX Firewall console access via Secure Shell (SSH). (Configuration mode.)

```

ssh disconnect session_id

no ssh disconnect session_id

ssh ip_address [netmask] [interface_name]

no ssh ip_address [netmask] [interface_name]

ssh timeout mm

no timeout mm

show ssh [sessions [ip_address]]

show ssh timeout

clear ssh

```

## Syntax Description

<i>ip_address</i>	IP address of the host or network authorized to initiate an SSH connection to the PIX Firewall.
<i>netmask</i>	Network mask for <i>ip_address</i> . If you do not specify a <i>netmask</i> , the default is 255.255.255.255 regardless of the class of <i>ip_address</i> .
<i>interface_name</i>	PIX Firewall interface name on which the host or network initiating the SSH connection resides.
<i>mm</i>	The duration in minutes that a session can be idle before being disconnected. The default duration is 5 minutes. The allowable range is from 1 to 60 minutes.
<i>session_id</i>	SSH session ID number available from the <b>show ssh sessions</b> command.

## Usage Guidelines

The **ssh ip\_address** command specifies the host or network authorized to initiate an SSH connection to the PIX Firewall. The **ssh timeout** command lets you specify the duration in minutes that a session can be idle before being disconnected. The default duration is 5 minutes. Use the **show ssh sessions** command to list all active SSH sessions on the PIX Firewall. The **ssh disconnect** command lets you disconnect a specific session you observed from the **show ssh sessions** command. Use the **clear ssh** command to remove all **ssh** command statements from the configuration. Use the **no ssh** command to remove selected **ssh** command statements from the configuration.



### Note

You must generate an RSA key-pair for the PIX Firewall before clients can connect to the PIX Firewall console. To use SSH, your PIX Firewall must have a DES or 3DES activation key.

To gain access to the PIX Firewall console via SSH, at the SSH client, enter the username as **pix** and enter the Telnet password. You can set the Telnet password with the **passwd** command; the default Telnet password is **cisco**. To authenticate using AAA server instead, configure the **aaa authenticate ssh console** command.

SSH permits up to 100 characters in a username and up to 50 characters in a password.

When starting an SSH session, a dot (.) displays on the PIX Firewall console before the SSH user authentication prompt appears.

The dot appears as follows:

```
pixfirewall(config)# .
pixfirewall(config)# .
```

The display of the dot does not affect the functionality of SSH. The dot appears on at the console when generating a server key or decrypting a message using private keys during SSH key exchange, before user authentication occurs. These tasks can take up to two minutes or longer. The dot is a progress indicator that verifies that the PIX Firewall is busy and has not hung.

### show ssh sessions Command

The **show ssh sessions** command provides the following display:

Session ID	Client IP	Version	Encryption	State	Username
0	172.16.25.15	1.5	3DES	4	-
1	172.16.38.112	1.5	DES	6	pix
2	172.16.25.11	1.5	3DES	4	-

The Session ID is a unique number that identifies an SSH session. The Client IP is the IP address of the system running an SSH client. The Version lists the version number of the SSH client. The Encryption column lists the type of encryption the SSH client is using. The State column lists the progress the client is making as it interacts with the PIX Firewall. The Username column lists the username that the client is accessing. The “pix” username appears when an SSH client is accessing the PIX Firewall console.

The following table lists the SSH states that appear in the State column:

Number	SSH State
0	SSH_CLOSED
1	SSH_OPEN
2	SSH_VERSION_OK
3	SSH_SESSION_KEY_RECEIVED
4	SSH_KEYS_EXCHANGED
5	SSH_AUTHENTICATED
6	SSH_SESSION_OPEN
7	SSH_TERMINATE
8	SSH_SESSION_DISCONNECTING
9	SSH_SESSION_DISCONNECTED
10	SSH_SESSION_CLOSED

### SSH Syslog Messages

Syslog messages 315001, 315002, 315003, 315004, and 315011 were added for SSH. Refer to *System Log Messages for the Cisco Secure PIX Firewall Version 5.2* for more information.

### Obtaining an SSH Client

The following sites let you download an SSH v1.x client. Because SSH version 1.x and 2 are entirely different protocols and are not compatible, be sure you download a client that supports SSH v1.x.

- Windows 3.1, Windows CE, Windows 95, and Windows NT 4.0—download the free Tera Term Pro SSH v1.x client from the following site:

<http://hp.vector.co.jp/authors/VA002416/teraterm.html>

The TTSSH security enhancement for Tera Term Pro is available at the following site:

<http://www.zip.com.au/~roca/ttssh.html>



**Note** You must download TTSSH to use Tera Term Pro with SSH. TTSSH provides a Zip file you copy to your system. Extract the zipped files into the same folder that you installed Tera Term Pro. For a Windows 95 system, by default, this would be the C:\Program Files\Ttempro folder.

- Linux, Solaris, OpenBSD, AIX, IRIX, HP/UX, FreeBSD, and NetBSD—download the SSH v1.x client from the following site:

<http://www.openssh.com>

- Macintosh (international users only)—download the Nifty Telnet 1.1 SSH client from the following site:

<http://www.lysator.liu.se/~jonasw/freeware/niftyssh/>

### Changed **aaa** Command for SSH

The **aaa** command adds the **ssh** option for use with SSH:

```
aaa authentication [serial | enable | telnet | ssh] console group_tag
```

The new **ssh** option specifies the group of AAA servers to be used for SSH user authentication. The authentication protocol and AAA server IP addresses are defined with the **aaa-server** command statement.

Similar to the Telnet model, if an **aaa authentication ssh console group\_tag** command statement is not defined, you can gain access to the PIX Firewall console with the username **pix** and with the PIX Firewall Telnet password (set with the **passwd** command). If the **aaa** command is defined, but the SSH authentication request times out, this implies that the AAA server may be down or not available. You can gain access to the PIX Firewall using the username **pix** and the enable password (set with the **enable password** command). By default, the Telnet password is **cisco** and the enable password is not set. If the enable password is empty (null), even if you enter the password correctly, you are not granted access to the SSH session.

The user authentication attempt limit is set to 3. Note that the Linux version of the SSH version 1 client available from <http://www.openssh.com> only allows one user authentication attempt.

See also: **aaa**, **ca**, **domain-name**, **enable password**, **hostname**, **passwd**. The **ca** and **domain-name** commands are described in the *IPSec User Guide for the Cisco Secure PIX Firewall Version 5.2*.

### Examples

Create an RSA key-pair with a modulus size of 1024 bits (recommended for use with Cisco IOS software):

```
hostname cisco-pix
domain-name example.com
ca generate rsa key 1024
show ca mypubkey rsa
ca save all
```

These command statements set the hostname and domain name for the PIX Firewall, generate the RSA key-pair, display the RSA key-pair, and save the RSA key-pair to Flash memory.

Start an SSH session so clients on the outside interface can access the PIX Firewall console remotely over a secure shell:

```
ssh 10.1.1.1 255.255.255.255 outside
ssh timeout 60
```

Configure the PIX Firewall to perform user authentication using AAA servers. The protocol is the protocol used by the AAA-server to do the authentication. The following example uses the TACACS+ authentication protocol:

```
aaa-server ssh123 (inside) host 10.1.1.200 mysecure
aaa-server ssh123 protocol tacacs+
aaa authenticate ssh console ssh123
```

# static

Map local IP address to a global IP address. (Configuration mode.)

```
static [(internal_if_name, external_if_name)] global_ip local_ip [netmask network_mask]
[max_conns [em_limit]] [norandomseq]
```

```
no static [(internal_if_name, external_if_name)] global_ip local_ip [netmask network_mask]
[max_conns [em_limit]] [norandomseq]
```

```
show static
```

## Syntax Description

<i>internal_if_name</i>	The internal network interface name. The higher security level interface you are accessing.
<i>external_if_name</i>	The external network interface name. The lower security level interface you are accessing.
<i>global_ip</i>	A global IP address. This address cannot be a PAT (Port Address Translation) IP address. The IP address on the lower security level interface you are accessing.
<i>local_ip</i>	The local IP address from the inside network. The IP address on the higher security level interface you are accessing.
<b>netmask</b>	Reserve word required before specifying the network mask.
<i>network_mask</i>	The network mask pertains to both <i>global_ip</i> and <i>local_ip</i> . For host addresses, always use 255.255.255.255. For network addresses, use the appropriate class mask or subnet mask; for example, for Class A networks, use 255.0.0.0. An example subnet mask is 255.255.255.224.
<i>max_conns</i>	The maximum number of connections permitted through the static at the same time.
<i>em_limit</i>	The embryonic connection limit. An embryonic connection is one that has started but not yet completed. Set this limit to prevent attack by a flood of embryonic connections. The default is 0, which means unlimited connections.
<b>norandomseq</b>	Do not randomize the TCP/IP packet's sequence number. Only use this option if another inline firewall is also randomizing sequence numbers and the result is scrambling the data. Use of this option opens a security hole in the PIX Firewall.

## Usage Guidelines

The **static** command creates a permanent mapping (called a static translation slot or "xlate") between a local IP address and a global IP address. Use the **static** and **access-list** commands when you are accessing an interface of a higher security level from an interface of a lower security level; for example, when accessing the inside from a perimeter or the outside interface.

### TCP Intercept Feature

Prior to version 5.2, PIX Firewall offered no mechanism to protect systems reachable via a static and TCP conduit from TCP SYN attacks. Previously, if an embryonic connection limit was configured in a **static** command statement, PIX Firewall simply dropped new connection attempts once the embryonic threshold was reached. Given this, a modest attack could stop an institution's Web traffic. For **static** command statements without an embryonic connection limit, PIX Firewall passes all traffic. If the affected system does not have TCP SYN attack protection, and most operating systems do not offer sufficient protection, then the affected system's embryonic connection table overloads and all traffic stops.

With the new TCP intercept feature, once the optional embryonic connection limit is reached, and until the embryonic connection count falls below this threshold, every SYN bound for the affected server is intercepted. For each SYN, PIX Firewall responds on behalf of the server with an empty SYN/ACK segment. PIX Firewall retains pertinent state information, drops the packet, and waits for the client's acknowledgement. If the ACK is received, then a copy of the client's SYN segment is sent to the server and the TCP three-way handshake is performed between PIX Firewall and the server. If and only if, this three-way handshake completes, may the connection resume as normal. If the client does not respond during any part of the connection phase, then PIX Firewall retransmits the necessary segment using exponential back-offs.

This feature requires no change to the PIX Firewall command set, only that the embryonic connection limit on the **static** command now has a new behavior.

### Deny Xlate for Network or Broadcast Address for Inbound Traffic

For all inbound traffic, PIX Firewall denies translations for destination IP addresses identified as network address or broadcast addresses. PIX Firewall utilizes the global IP and mask from a **static** command statement to differentiate regular IP addresses from network or broadcast addresses. If a global IP address is a valid network address with a matching network mask, then PIX Firewall disallows the xlate for network or broadcast IP addresses with inbound packet.

### Interface Names

The interface names on the **static** command may seem confusing at first. This is further complicated by how NAT is handled on the PIX Firewall. If NAT is disabled, with the **nat 0** command, statics are specified with a different set of rules than when NAT is enabled. For either no NAT or NAT, the rule of which command to access an interface stays the same as shown in Table 5-9.

Table 5-9 assumes that the security levels are 40 for dmz1 and 60 for dmz2.

**Table 5-9** *Interface Access Commands by Interface*

From This Interface	To This Interface	Use This Command
inside	outside	<b>nat</b>
inside	dmz1	<b>nat</b>
inside	dmz2	<b>nat</b>
dmz1	outside	<b>nat</b>
dmz1	dmz2	<b>static</b>
dmz1	inside	<b>static</b>
dmz2	outside	<b>nat</b>
dmz2	dmz1	<b>nat</b>

**Table 5-9 Interface Access Commands by Interface (continued)**

From This Interface	To This Interface	Use This Command
dmz2	inside	<b>static</b>
outside	dmz1	<b>static</b>
outside	dmz2	<b>static</b>
outside	inside	<b>static</b>

**With NAT Enabled**

NAT (Network Address Translation) is enabled with the **nat n** command where “n” has the value **1** or greater; for example, **nat 1 0 0**.

Always specify the interface name of the highest security level interface you are accessing, followed by the lower security level interface. The IP addresses are also confusing because the first IP address you specify is for the lower security level interface. The second IP address is for the higher security level interface. The way to remember this is as follows:

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

For example, assume you have four interfaces on the PIX Firewall that have security levels set with the **nameif** command as follows:

```
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 dmz1 security40
nameif ethernet3 dmz2 security60
```

To access the inside from the outside interface, you need a **static** command like the following:

```
static (inside,outside) outside_ip_address inside_ip_address netmask mask
```

Replace *outside\_ip\_address* with the global IP address (an IP address on the lower security level interface). Replace *inside\_ip\_address* with the IP address of the host on the higher security level interface that you want to grant access to.

Use these replacements in the rest of the commands in this section. Replace *mask* with 255.255.255.255 for host addresses, except when subnetting is in effect; for example, 255.255.255.128. For network addresses, use the appropriate class mask; for example, for Class A networks, use 255.0.0.0.

To access the inside from the dmz1 interface, you need a **static** command like the following:

```
static (inside,dmz1) dmz1_ip_address inside_ip_address netmask mask
```

To access the inside from the dmz2 interface, you need a **static** command like the following:

```
static (inside,dmz2) dmz2_ip_address inside_ip_address netmask mask
```

To access the dmz2 interface from the dmz1 interface, you need a **static** command like the following:

```
static (dmz2,dmz1) dmz1_ip_address dmz2_ip_address netmask mask
```

To go the other way around, from a higher security level interface to a lower security level interface, use the **nat** and **global** commands. For example, to access dmz1 from dmz2, use these commands:

```
nat (dmz2) 1 0 0
global (dmz1) 1 global_ip_address-global_ip_address
```

Replace *global\_ip\_address-global\_ip\_address* with the IP address range of the addresses in the pool of global addresses. The **nat** command specifies the name of the higher security level interface; the pool of global addresses are on the lower security level interface.

View the **nat** command page for more information on using these commands.

**Note**

If you use a **static** command, you must also use an **access-list** command. The **static** command makes the mapping, the **access-list** command lets users access the **static** mapping.

The first IP address you specify in the **static** command is the first IP address you specify in the **access-list** command as shown in this example:

```
static (dmz2,dmz1) 10.1.1.1 192.168.1.1 netmask 255.255.255.255
access-list acl_dmz1 permit tcp 10.1.1.0 255.255.255.0 host 10.1.1.1
access-group acl_dmz1 in interface dmz1
```

The **static** command maps the address 10.1.1.1 on the dmz1 interface so that users on the dmz1 interface can access the 192.168.1.1 host on the dmz2 interface. The **access-list** command lets any users in the 10.1.1.0 network access the 10.1.1.1 address over any TCP port. The **access-group** command statement binds the **access-list** command statement to the dmz1 interface.

**Note**

Always make **access-list** command statements as specific as possible. Using the **any** option to allow any host access should be used with caution for access lists used with statics.

**With No-NAT**

With no-NAT, the **static** command has a different sense of logic. With NAT disabled, addresses on both sides of the firewall are registered addresses. Between interfaces, addresses must be on different subnets that you control with subnetting. See Appendix D, “Subnet Masking and Addressing,” for more information.

Without address translation, you protect addresses on the inside or perimeter interfaces by not providing access to them. Without an **access-list** command statement, the inside host cannot be accessed on the outside and is, in effect, invisible to the outside world. Conversely, only by opening statics and access lists to servers on the inside or perimeter interfaces, do the hosts become visible.

Without address translation, the format of the **static** command becomes different:

```
static (high,low) high high
```

Again, the security level set for each interface with the **nameif** command determines what information you fill in. You are using **static** to access a higher security interface from a lower security interface. The IP address you want visible on the lower security interface is that of the higher security interface. This is the IP address users on the lower security interface’s network will use to access the server on the higher security level interface’s network. Because address translation is not occurring, the actual address of the server is presented as both the visible address and the address of the host.

For example, a web server on the dmz, 209.165.201.5 needs to be accessible by users on the outside. The **static** and **access-list** command statements are as follows:

```
static (dmz,outside) 209.165.201.5 209.165.201.5 netmask 255.255.255.255
access-list acl_out permit tcp any host 209.165.201.5 eq www
access-group acl_out in interface outside
```

The **static** command presents the 209.165.201.5 address on the outside interface. The DNS server on the outside would map this IP address to the domain of the company; for example, example.com. Users accessing example.com are permitted to access the web server via port 80 by the **access-list** command.

Another example of no-NAT statics would be when users on dmz1 need to access a web server on dmz2. The network uses a Class C address and subnets it with the .240 subnet. Addresses 209.165.201.1 to 209.165.201.14 are on dmz1, and addresses 209.165.201.17 to 209.165.201.30 are on dmz2. The web server is at 209.165.201.25. The **static** and **access-list** command statements are as follows:

```
static (dmz2,dmz1) 209.165.201.25 209.165.201.25 netmask 255.255.255.255
access-list acl_dmz1 permit tcp any host 209.165.201.25 eq www
access-group acl_dmz1 in interface dmz1
```

The **static** command statement opens access to the web server at 209.165.201.25. The **access-list** command statement permits access to the web server only on port 80 (**www**).

#### Additional static Information

After changing or removing a **static** command statement, use the **clear xlate** command.

You can create a single mapping between the global and local hosts, or create a range of statics known as net statics.

The **static** command determines the network mask of network statics by the **netmask** option or by the number in the first octet of the global IP address. The **netmask** option can be used to override the number in the first octet. If the address is all zeros where the net mask is zero, then the address is a net address.



#### Note

---

Do not create statics with overlapping global IP addresses.

---

See also: **access-list**

#### Examples

The example that follows creates a **static** command and then permits users to call in through H.323 using Intel InternetPhone, CU-SeeMe, CU-SeeMe Pro, MeetingPoint, or MS NetMeeting to 10.1.1.2 using IP address 209.165.201.2, to 10.1.1.10 using IP address 209.165.201.10, and so on. The net **static** command that follows maps addresses 209.165.201.1 through 209.165.201.30 to local addresses 10.1.1.1 through 10.1.1.30.

```
static (inside, outside) 209.165.201.0 10.1.1.0 netmask 255.255.255.255
access-list acl_out permit tcp any 209.165.201.0 255.255.255.224 eq h323
access-group acl_out in interface outside
```

The following example shows the commands used to disable Mail Guard:

```
static (dmz1,outside) 209.165.201.1 10.1.1.1 netmask 255.255.255.255
access-list acl_out permit tcp any host 209.165.201.1 eq smtp
access-group acl_out in interface outside
no fixup protocol smtp 25
```

In this example, the **static** command sets up a global address to permit outside hosts access to the 10.1.1.1 mail server host on the dmz1 interface. (The MX record for DNS needs to point to the 209.165.201.1 address so that mail is sent to this address.) The **access-list** command lets any outside users access the global address through the SMTP port (25). The **no fixup protocol** command disables the Mail Guard feature.

# syslog

Enable syslog message facility. Obsolete command replaced by the **logging** command. (Privileged mode.)

**Note**

---

See the **logging** command for more information. The **syslog** command is available for backward compatibility.

---

# sysopt

Change PIX Firewall system options. (Configuration mode.)

```
sysopt connection enforcesubnet
no sysopt connection enforcesubnet
```

```
sysopt connection permit-pptp
no sysopt connection permit-pptp
```

```
sysopt connection tcpmss bytes
no sysopt connection tcpmss bytes
```

```
sysopt connection timewait
no sysopt connection timewait
```

```
sysopt nodnsalias inbound
sysopt nodnsalias outbound
no sysopt nodnsalias inbound
no sysopt nodnsalias outbound
```

```
sysopt noproxyarp if_name
no sysopt noproxyarp if_name
```

```
sysopt security fragguard
no sysopt security fragguard
```

```
sysopt radius ignore-secret
no sysopt radius ignore-secret
```

```
sysopt route dnat
no sysopt route dnat
```

```
clear sysopt
```

```
show sysopt
```

## Syntax Description

<b>connection enforcesubnet</b>	Enable spoof address filtering based on subnet.
<b>connection permit-pptp</b>	Allow PPTP traffic to bypass <b>conduit</b> or <b>access-list</b> command statement checking.
<b>connection tcpmss <i>bytes</i></b>	Force TCP proxy connection to have a maximum segment size no greater than <i>bytes</i> . The default value for bytes is 1380.
<b>connection timewait</b>	Force each TCP connection to linger in a shortened TIME_WAIT state of at least 15 seconds after the final normal TCP close-down sequence.

<b>nodnsalias inbound</b>	Disable inbound embedded DNS A record fixups according to aliases that apply to the A record address.
<b>nodnsalias outbound</b>	Disable outbound DNS A record replies.
<b>noproxyarp</b> <i>if_name</i>	Disable proxy-arps on a PIX Firewall interface.
<b>route dnat</b>	Specify that when an incoming packet does a route lookup, the incoming interface is used to determine which interface the packet should go to, and which is the next hop.
<b>security fragguard</b>	Enable the IP Frag Guard feature.
<b>radius ignore-secret</b>	Ignore authenticator key to avoid retransmit caveat.

### Usage Guidelines

The **sysopt** commands let you tune various PIX Firewall security and configuration features. In addition, you can use this command to disable the PIX Firewall IP Frag Guard feature.

For information on the IPSec-related **sysopt** commands, refer to the **sysopt** command page within the “Command Reference” chapter of the *IPSec User Guide for the Cisco Secure PIX Firewall Version 5.2*.

### **sysopt connection enforcesubnet**

The **sysopt connection enforcesubnet** command prevents external users from spoofing internal addresses. This command prevents packets with a source address belonging to the destination subnet from traversing the PIX Firewall. For example, if a packet arrives from the outside but has a source address belonging to the inside subnet, the PIX Firewall does not let the packet through.

The **sysopt connection enforcesubnet** command applies only to inbound connections.

To configure the PIX Firewall to detect spoofed IP addresses, use explicit **access-list deny** command statements in the configuration; for example:

```
access-list acl_grp deny ip any in_host_net1 in_host_net1_mask
access-list acl_grp deny ip any in_host_net2 in_host_net2_mask
```

Replace *in\_host\_netn* with the addresses on the internal network.

### **sysopt connection permit-pptp**

Let PPTP traffic bypass **conduit** and **access-list** command statement checking. Use the **vpdn** command to implement PPTP.

### Examples

In the following example, a PPTP client authenticates using **mschap**, negotiates **mppe** encryption, receives the **dns** and **wins** server addresses, and Telnets to the host 192.168.0.2 directly through the **nat 0** command.

```
ip local pool my-addr-pool 10.1.1.1-10.1.1.254
aaa-server my-aaa-server-group (inside) host 192.168.0.10 key 12345678
aaa-server my-aaa-server-group protocol radius
vpdn group 1 accept dialin pptp
vpdn group 1 ppp authentication mschap
vpdn group 1 ppp encryption mppe auto required
vpdn group 1 client configuration address local my-addr-pool
vpdn group 1 client authentication aaa my-aaa-server-group
vpdn group 1 client configuration dns 10.2.2.99
vpdn group 1 client configuration wins 10.2.2.100
vpdn enable outside
access-list nonat permit ip 10.1.1.0 255.255.255.0 host 192.168.0.2
access-list nonat permit ip 10.1.1.0 255.255.255.0 host 10.2.2.99
access-list nonat permit ip 10.1.1.0 255.255.255.0 host 10.2.2.100
nat (inside) 0 access-list nonat
sysopt connection permit-pptp
```

### sysopt connection tcpmss

The **sysopt connection tcpmss** command forces proxy TCP connections to have a maximum segment size no greater than *bytes*. This command requests that each side not send a packet of a size greater than *bytes* at any time during the initial TCP connection establishment.



#### Note

If the client sending the proxy TCP connection does not announce a maximum segment size, PIX Firewall assumes that the RFC 793 default value of 536 bytes is in effect. If the client announces a maximum segment size larger than the number of *bytes*, PIX Firewall reduces the maximum segment size to *bytes*.

The *bytes* value can be a minimum of 28 and any maximum number. You can disable this feature by setting *bytes* to zero. By default, the PIX Firewall sets 1380 bytes as the **sysopt connection tcpmss** even though this command does not appear in the default configuration. The calculation for setting the TCP maximum segment size to 1380 bytes is as follows:

$$1380 \text{ data} + 20 \text{ TCP} + 20 \text{ IP} + 24 \text{ AH} + 24 \text{ ESP\_CIPHER} + 12 \text{ ESP\_AUTH} + 20 \text{ IP} = 1500 \text{ bytes}$$

1500 bytes is the MTU for Ethernet connections. Cisco recommends that the default value of 1380 bytes be used for Ethernet and mixed Ethernet and Token Ring environments. If the PIX Firewall has all Token Ring interfaces, you can set *bytes* to 4056. However, if even one link along the path through the network is not a Token Ring, setting *bytes* to such a high value may cause poor throughput. In its 1380 byte default value, this command increases throughput of the **sysopt security fragguard** command.

The TCP maximum segment size is the maximum size that an end host can inject into the network at one time (see RFC 793 for more information on the TCP protocol). The **sysopt connection tcpmss** command is recommended in a network environment being attacked being with overly aggressive TCP or HTTP stack with a faulty path MTU value that is degrading the performance of the PIX Firewall IP Frag Guard feature. Environments where one or more end hosts reside on a Token Ring network are especially susceptible to this faulty behavior.



#### Note

Although, not advised for normal use of this feature, if you encounter the syslog IPFRAG messages 209001 and 209002, you can raise the *bytes* value.

**sysopt connection timewait**

The **sysopt connection timewait** command is necessary for end host applications whose default TCP terminating sequence is a simultaneous close instead of the normal shutdown sequence (see RFC 793). In a simultaneous close, both ends of the transaction initiate the closing sequence, as opposed to the normal sequence where one end closes and the other end acknowledges prior to initiating its own closing sequence.

The default behavior of the PIX Firewall is to track the normal shutdown sequence and release the connection after two FINs and the ACKnowledgment of the last FIN segment. This quick release heuristic enables the PIX Firewall to sustain a high connection rate.

However with a simultaneous close, the quick release forces one side of the connection to linger in the CLOSING state (see RFC 793). Many sockets in the CLOSING state can degrade the performance of an end host. For instance, some WinSock mainframe clients are known to exhibit this behavior and degrade the performance of the mainframe server. Old versions of HP/UX are also susceptible to this behavior. Enabling the **sysopt connection timewait** command enables a quiet time window for the abnormal close down sequence to complete.

The **no sysopt connection timewait** command disables the option, which is off by default.

**Note**


---

Use of the **sysopt connection timewait** command may impact PIX Firewall performance especially with low memory configuration and highly dynamic traffic pattern such as HTTP.

---

**sysopt nodnsalias**

The **sysopt nodnsalias inbound** disables inbound embedded DNS A record fixups according to aliases that apply to the A record address. **sysopt nodnsalias outbound** affects outbound replies.

This command remedies the case when a DNS server is on the outside and users on the inside need to access a server on a perimeter interface. In the past, you would use the **alias** command to permit DNS responses to resolve correctly through the PIX Firewall, but formerly you had to reverse the parameters for the local IP address and foreign IP address.

For example, you would normally code the **alias** command as follows:

```
alias (inside) 192.168.1.4 209.165.201.11 255.255.255.255
```

Inside host 192.168.1.5 needs access to www.example.com, which resolves at an outside ISP DNS to 209.165.201.11. The PIX Firewall fixes this DNS response sending the host a response of 192.168.1.4. The host uses its gateway (the PIX Firewall) to go to 192.168.1.4, which the PIX Firewall now aliases back to the 209.165.201.11. Because this is actually 192.168.1.4, a server on the perimeter interface of the PIX Firewall, the packet is dropped because the PIX Firewall sent the packet to the outside interface, which is the incorrect interface.

The **sysopt nodnsalias inbound** command has the same effect as reversing the **alias** command statement parameters as follows:

```
alias (inside) 209.165.201.11 192.168.1.4 255.255.255.255
```

This works properly because everything happens in reverse. The DNS is now modified to 209.165.201.11 and the host inside uses its gateway (the PIX Firewall) to get there, the PIX Firewall aliases this back to 192.168.1.4 and routes it out the perimeter interface to the correct host and the TCP connection is established.

**sysopt noproxyarp**

The **sysopt noproxyarp** command lets you disable proxy-arps on a PIX Firewall interface.

**sysopt radius ignore-secret**

Some commonly used RADIUS servers, such as Livingston version 1.16, have a usage caveat where they do not include the key in the authenticator hash in the accounting acknowledgment response. This can cause the PIX Firewall to continually retransmit the accounting request. Use the **sysopt radius ignore-secret** command to cause the PIX Firewall to ignore the key in the authenticator of accounting acknowledgments thus avoiding the retransmit problem. (The key described here is the key you set with the **aaa-server** command.)

**sysopt route dnat**

The **sysopt route dnat** command specifies that when an incoming packet does a route lookup, the incoming interface is used to determine which interface the packet should go to, and which is the next hop.

**sysopt security fragguard**

The **sysopt security fragguard** command enables the IP Frag Guard feature. This feature is disabled by default. This feature enforces two additional security checks in addition to the security checks recommended by RFC 1858 against the many IP fragment style attacks: teardrop, land, and so on. First, each non-initial IP fragment is required to be associated with an already seen valid initial IP fragment. Second, IP fragments are rated to 100 full IP fragmented packets per second to each internal host.

The IP Frag Guard feature operates on all interfaces in the PIX Firewall and cannot be selectively enabled or disabled by interface.

PIX Firewall uses the **security fragguard** command to enforce the security policy determined by a **access-list permit** or **access-list deny** command to permit or deny packets through the PIX Firewall.

**Note**

Use of the **sysopt security fragguard** command breaks normal IP fragmentation conventions. However, not using this command exposes PIX Firewall to the possibility of IP fragmentation attacks. Cisco recommends that packet fragmentation not be permitted on the network if at all possible.

**Note**

If PIX Firewall is used as a tunnel for FDDI packets between routers, disable the **security fragguard** command feature.

**Note**

Because Linux sends IP fragments in reverse order, fragmented Linux packets will not pass through the PIX Firewall with the **sysopt security fragguard** command enabled.

The **show sysopt** command lists the **sysopt** commands in the configuration. The **clear sysopt** command resets the **sysopt** command to default settings. The **no sysopt security fragguard** command disables the IP Frag Guard feature.

**Examples**

The following example disables IP Frag Guard and then lists the current command options:

```
no sysopt security fragguard
show sysopt
sysopt security fragguard
no sysopt connection tcpmss
no sysopt connection timewait
```

# telnet

Specify host for PIX Firewall console access via Telnet. (Configuration mode.)

```
telnet ip_address [netmask] [if_name]  
clear telnet [ip_address [netmask] [if_name]]  
no telnet [ip_address [netmask] [if_name]]  
show telnet  
telnet timeout minutes  
show telnet timeout
```

## Syntax Description

<i>ip_address</i>	An IP address of a host or network that can access the PIX Firewall Telnet console. If an interface name is not specified, the address is assumed to be on an internal interface. PIX Firewall automatically verifies the IP address against the IP addresses specified by the <b>ip address</b> commands to ensure that the address you specify is on an internal interface. If an interface name is specified, PIX Firewall only checks the host against the interface you specify.
<i>netmask</i>	Bit mask of <i>ip_address</i> . To limit access to a single IP address, use 255 in each octet; for example, 255.255.255.255. If you do not specify <i>netmask</i> , it defaults to 255.255.255.255 regardless of the class of <i>local_ip</i> . Do not use the subnetwork mask of the internal network. The <i>netmask</i> is only a bit mask for the IP address in <i>ip_address</i> .
<i>if_name</i>	If IPSec is operating, PIX Firewall lets you specify an unsecure interface name, typically, the outside interface. At a minimum, the <b>crypto map</b> command must be configured to specify an interface name with the <b>telnet</b> command.
<b>timeout</b> <i>minutes</i>	The number of minutes that a Telnet session can be idle before being closed by PIX Firewall. The default is 5 minutes. The range is <b>1</b> to <b>60</b> minutes.

## Usage Guidelines

The **telnet** command lets you specify which hosts can access the PIX Firewall console with Telnet. You can enable Telnet to the PIX Firewall on all interfaces. However, the PIX Firewall enforces that all Telnet traffic to the outside interface be IPSec protected. Therefore, to enable Telnet session to the outside interface, configure IPSec on the outside interface to include IP traffic generated by the PIX Firewall and enable Telnet on the outside interface.

Up to 16 hosts or networks are allowed access to the PIX Firewall console with Telnet, 5 simultaneously. The **show telnet** command displays the current list of IP addresses authorized to access the PIX Firewall. Use the **no telnet** or **clear telnet** command to remove Telnet access from a previously set IP address. Use the **telnet timeout** feature to set the maximum time a console Telnet session can be idle before being logged off by PIX Firewall. The **clear telnet** command does not affect the **telnet timeout** command duration. The **no telnet** command cannot be used with the **telnet timeout** command.

Use the **passwd** command to set a password for Telnet access to the console. The default is **cisco**. Use the **who** command to view which IP addresses are currently accessing the firewall console. Use the **kill** command to terminate an active Telnet console session.

If the **aaa** command is used with the **console** option, Telnet console access must be authenticated with an authentication server.



#### Note

If you have configured the **aaa** command to require authentication for PIX Firewall Telnet console access and the console login request times out, you can gain access to the PIX Firewall from the serial console by entering the **pix** username and the password that was set with the **enable password** command.

#### Usage Notes

1. If you do not specify the interface name, the **telnet** command adds command statements to the configuration to let the host or network access the Telnet console from all internal interfaces. When you use the **show telnet** command, this assumption may not seem to make sense. For example, if you enter the following command without a netmask or interface name:

```
telnet 192.168.1.1
```

If you then use the **show telnet** command, you see that not just one command statement is specified, but all internal interfaces are represented with a command statement:

```
show telnet
192.168.1.1 255.255.255.255 inside
192.168.1.1 255.255.255.255 intf2
192.168.1.1 255.255.255.255 intf3
```

The purpose of the **show telnet** command is that, were it possible, the 192.168.1.1 host could access the Telnet console from any of these internal interfaces. An additional facet of this behavior is that you have to delete each of these command statements individually with the following commands:

```
no telnet 192.168.1.1 255.255.255.255 inside
no telnet 192.168.1.1 255.255.255.255 intf2
no telnet 192.168.1.1 255.255.255.255 intf3
```

2. To access the PIX Firewall with Telnet from the intf2 perimeter interface, use the following command:
 

```
telnet 192.168.1.1 255.255.255.255 intf2
```
3. The default password to access the PIX Firewall console via Telnet is **cisco**.
4. Some Telnet applications such as the Windows 95 or Windows NT Telnet sessions may not support access to the PIX Firewall unit's command history feature via the arrow keys. However, you can access the last entered command by pressing Ctrl-P.
5. The **telnet timeout** command affects the next session started but not the current session.
6. If you connect a computer directly to the inside interface of the PIX Firewall with Ethernet to test Telnet access, you must use a cross-over cable and the computer must have an IP address on the same subnet as the inside interface. The computer must also have its default route set to be the inside interface of the PIX Firewall.
7. Telnet access to the console must be configured before you use PIX Firewall Manager.

8. If you need to access the PIX Firewall console from outside the PIX Firewall, you can use a **static** and **access-list** command pair to permit a Telnet session to a Telnet server on the inside interface, and then from the server to the PIX Firewall. In addition, you can attach the console port to a modem but this may add a security problem of its own. You can use the same terminal settings as for HyperTerminal, which is described in Chapter 2, “Configuring the PIX Firewall.”

If you have IPSec configured, you can access the PIX Firewall console with Telnet from outside the PIX Firewall. Once an IPSec tunnel is created from an outside host to the PIX Firewall, you can access the console from the outside host.

9. Output from the **debug crypto ipsec**, **debug crypto isakmp**, and **debug ssh** commands do not display in a Telnet or SSH console session. For information about the **debug crypto ipsec** and **debug crypto isakmp** commands, refer to the **debug** command page within the “Command Reference” chapter of the *IPSec User Guide for the Cisco Secure PIX Firewall Version 5.2*.

See also: **aaa**, **kill**, **passwd**, **who**.

### Examples

The following examples permit hosts 192.168.1.3 and 192.168.1.4 to access the PIX Firewall console via Telnet. In addition, all the hosts on the 192.168.2.0 network are given access:

```
telnet 192.168.1.3 255.255.255.255 inside
telnet 192.168.1.4 255.255.255.255 inside
telnet 192.168.2.0 255.255.255.0 inside
show telnet
    192.168.1.3 255.255.255.255 inside
    192.168.1.4 255.255.255.255 inside
    192.168.2.0 255.255.255.0 inside
```

You can remove individual entries with the **no telnet** command or all **telnet** command statements with the **clear telnet** command:

```
no telnet 192.168.1.3 255.255.255.255 inside
show telnet
    192.168.1.4 255.255.255.255 inside
    192.168.2.0 255.255.255.0 inside
clear telnet
show telnet
```

You can change the maximum session idle duration as follows:

```
telnet timeout 10
show telnet timeout
telnet timeout 10 minutes
```

An example Telnet console login session appears as follows (the password does not display when entered):

```
PIX passwd: cisco

Welcome to the PIX Firewall
...
Type help or '?' for a list of available commands.
pixfirewall>
```

# terminal

Change console terminal settings. (Configuration mode.)

**terminal [no] monitor**

**terminal width** *characters*

## Syntax Description

**monitor** Enable or disable syslog message displays on the console.

**width** Set the width for displaying information during console sessions.

*characters* Permissible values are 0, which means 511 characters, or a value in the range of 40 to 511.

## Usage Guidelines

The **terminal monitor** command lets you enable or disable the display of syslog messages in the current session for either Telnet or serial access to the PIX Firewall console. Use the **logging monitor** command to enable or disable various levels of syslog messages to the console; use the **terminal no monitor** command to disable the messages on a per session basis. Use **terminal monitor** to restart the syslog messages for the current session.

The **terminal width** command sets the width for displaying command output. The terminal width is controlled by the command: **terminal width nn**, where *nn* is the width in characters. If you enter a line break, it is not possible to backspace to the previous line.

## Examples

The following example shows enabling logging and then disabling logging only in the current session with the **terminal no monitor** command:

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

# tftp-server

Specify the IP address of the TFTP configuration server. (Configuration mode.)

```
tftp-server [if_name] ip_address path  
no tftp-server [[if_name] ip_address path]  
clear tftp-server [[if_name] ip_address path]  
show tftp-server
```

## Syntax Description

<i>if_name</i>	Interface name on which the TFTP server resides. If not specified, an internal interface is assumed. If you specify the outside interface, a warning message informs you that the outside interface is unsecure.
<i>ip_address</i>	The IP address or network of the TFTP server.
<i>path</i>	The path and filename of the configuration file. The format for path differs by the type of operating system on the server. The contents of path are passed directly to the server without interpretation or checking. The configuration file must exist on the TFTP server. Many TFTP servers require the configuration file to be world-writable to write to it and world-readable to read from it.

## Usage Guidelines

The **tftp-server** command lets you specify the IP address of the server that you use to propagate PIX Firewall configuration files to your firewalls. Use the **tftp-server** command with the **configure net** command to read from the configuration or with the **write net** command to store the configuration in the file you specify. The **clear tftp-server** command removes the **tftp-server** command from your configuration.

PIX Firewall supports only one TFTP server.

The *path* name you specify in the **tftp-server** is appended to the end of the IP address you specify in the **configure net** and **write net** commands. The more you specify of a file and path name with the **tftp-server** command, the less you need to specify with the **configure net** and **write net** commands. If you specify the full path and filename in the **tftp-server** command, the IP address in the **configure net** and **write net** commands can be represented with a colon (:).

The **no tftp server** command disables access to the server. The **show tftp-server** command lists the **tftp-server** command statements in the current configuration.

## Examples

The following example specifies a TFTP server and then reads the configuration from /pixfirewall/config/test\_config:

```
tftp-server 10.1.1.42 /pixfirewall/config/test_config  
...  
configure net :
```

# timeout

Set the maximum idle time duration. (Configuration mode.)

```
timeout [xlate hh:mm:ss] [conn hh:mm:ss] [half-closed hh:mm:ss] [udp hh:mm:ss]
  [rpc hh:mm:ss] [h323 hh:mm:ss] [sip hh:mm:ss] [sip_media hh:mm:ss][uauth
  hh:mm:ss] [absolute | inactivity]
```

**clear timeout**

**show timeout**

## Syntax Description

<b>xlate</b> <i>hh:mm:ss</i>	Idle time until a translation slot is freed. This duration must be at least 1 minute. The default is 3 hours.
<b>conn</b> <i>hh:mm:ss</i>	Idle time until a connection slot is freed. Use <b>0:0:0</b> for the time value to never time out a connection. This duration must be at least 5 minutes. The default is 1 hour.
<b>half-closed</b> <i>hh:mm:ss</i>	Idle time until a TCP half-close connection is freed. The default is 10 minutes. Use <b>0:0:0</b> to never time out a half-closed connection. The minimum is 5 minutes.
<b>udp</b> <i>hh:mm:ss</i>	Idle time until a UDP slot is freed. This duration must be at least 1 minute. The default is 2 minutes.
<b>rpc</b> <i>hh:mm:ss</i>	Idle time until an RPC slot is freed. This duration must be at least 1 minute. The default is 10 minutes.
<b>sip</b> <i>hh:mm:ss</i>	Modifies the SIP timer. SIP signalling port is set to a default of 30 minutes.
<b>sip_media</b> <i>hh:mm:ss</i>	Modifies the media timer, which is used for SIP RTP/RTCP with SIP UDP media packets, instead of the UDP inactivity timeout. SIP media port is set to 2 minutes in the list of protocol timers.
<b>h323</b> <i>hh:mm:ss</i>	Duration for H.323 inactivity timer. When this time elapses, the port used by the H.323 service closes. This duration must be at least 5 minutes. The default is 5 minutes.
<b>uauth</b> <i>hh:mm:ss</i>	Duration before authentication and authorization cache times out and user has to reauthenticate next connection. This duration must be shorter than the <b>xlate</b> values. Set to <b>0</b> to disable caching. Do not set to zero if passive FTP is used on the connections.
<b>absolute</b>	Run <b>uauth</b> timer continuously, but after timer elapses, wait to reprompt the user until the user starts a new connection, such as clicking a link in a web browser. The default <b>uauth</b> timer is <b>absolute</b> . To disable <b>absolute</b> , set the <b>uauth</b> timer to <b>0</b> (zero).
<b>inactivity</b>	Start <b>uauth</b> timer after a connection becomes idle.

### Usage Guidelines

The **timeout** command sets the idle time for connection, translation UDP, RPC, and H.323 slots. If the slot has not been used for the idle time specified, the resource is returned to the free pool. TCP connection slots are freed approximately 60 seconds after a normal connection close sequence.

The **clear timeout** command sets the durations to their default values.

**Note**

---

Do not use the **timeout uauth 0:0:0** command if passive FTP is used for the connection, or if the **virtual** command is used for Web authentication.

---

**Note**

---

The connection timer takes precedence over the translation timer, such that the translation timer only works after all connections have timed out.

---

### uauth inactivity and absolute Qualifiers

The **uauth inactivity** and **absolute** qualifiers cause users to have to reauthenticate after either a period of inactivity or an absolute duration.

**Note**

---

If you set the inactivity timer to a duration, but the absolute timer to zero, then users are only reauthenticated after the inactivity timer elapses. If you set both timers to zero, then users have to reauthenticate on every new connection.

---

The inactivity timer starts after a connection becomes idle. If a user establishes a new connection before the duration of the inactivity timer, the user is not required to reauthenticate. If a user establishes a new connection after the inactivity timer expires, the user must reauthenticate. The default durations are zero for the inactivity timer and 5 minutes for the absolute timer; that is, the default behavior is to cause the user to reauthenticate every 5 minutes.

The absolute timer runs continuously, but waits to reprompt the user when the user starts a new connection, such as clicking a link and the absolute timer has elapsed, then the user is prompted to reauthenticate. The absolute timer must be shorter than the **xlate** timer; otherwise, a user could be reprompted after their session already ended.

Inactivity timers give users the best Web access because they are not prompted to regularly reauthenticate. Absolute timers provide security and manage the PIX Firewall connections better. By being prompted to reauthenticate regularly, users manage their use of the resources more efficiently. Also by being reprompted, you minimize the risk that someone will attempt to use another user's access after they leave their workstation, such as in a college computer lab. You may want to set an absolute timer during peak hours and an inactivity timer thereafter.

Both an inactivity timer and an absolute timer can operate at the same time, but you should set the absolute timer duration longer than the inactivity timer. If the absolute timer is less than the inactivity timer, the inactivity timer never occurs. For example, if you set the absolute timer to 10 minutes and the inactivity timer to an hour, the absolute timer reprompts the user every 10 minutes; therefore, the inactivity timer will never be started.

Use the **show timeout** command to display the current **timeout** command settings.

See also: **show xlate**, **uauth**.

**Note**

---

RPC and NFS are very unsecure protocols and should be used with caution.

---

### Examples

The following is sample output for the **show timeout** command:

```
show timeout  
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
```

The following is sample output for the **timeout** command in which variables are changed and then displayed with the **show timeout** command:

```
timeout uauth 0:5:00 absolute uauth 0:4:00 inactivity  
show timeout  
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 uauth 0:04:00 inactivity
```

# uauth (clear and show)

Delete all authorization caches for a user. (Privileged mode.)

```
clear uauth [username]
```

```
show uauth [username]
```

## Syntax Description

*username* Clear or view user authentication information by username.

## Usage Guidelines

The **clear uauth** command deletes one user's or all users' AAA authorization caches, which forces the user or users to reauthenticate the next time they create a connection. The **show uauth** command displays one or all currently authenticated users, the host IP to which they are bound, and, if applicable, any cached IP and port authorization information.

The **show uauth** command also lists CiscoSecure 2.1 and later idletime and timeout values, which can be set for different user groups.

Each user host's IP address has an authorization cache attached to it. If the user attempts to access a service that has been cached from the correct host, the firewall considers it preauthorized and immediately unproxies the connection. This means that once you are authorized to access a web site, for example, the authorization server is not contacted for each of the images as they are loaded (assuming they come from the same IP address). This significantly increases performance and reduces load on the authorization server.

The cache allows up to 16 address and service pairs for each user host.

The output from the **show uauth** command displays the username provided to the authorization server for authentication and authorization purposes, the IP address that the username is bound to, and whether the user is authenticated only, or has cached services.

Use the **timeout uauth** command to specify how long the cache should be kept after the user connections become idle. The **timeout** command value must be at least 2 minutes. Use the **clear uauth** command to delete all authorization caches for all users, which will cause them to have to reauthenticate the next time they create a connection.

See also: **aaa authorization**, **timeout**.

### Examples

The following is sample output for the **show uauth** command:

```
show uauth
user 'pat' from 209.165.201.2 authenticated
user 'robin' from 209.165.201.4 authorized to:
  port 192.168.67.34/telnet    192.168.67.11/http    192.168.67.33/tcp/8001
    192.168.67.56/tcp/25    192.168.67.42/ftp
user 'terry' from 209.165.201.7 authorized to:
  port 192.168.1.50/http    209.165.201.8/http
```

In this example, Pat has authenticated with the server but has not completed authorization. Robin has preauthorized connections to the Telnet, Web (HTTP), sendmail, FTP services, and to TCP port 8001 on 192.168.67.33.

Terry has been browsing the Web and is authorized for Web browsing to the two sites shown.

The next example causes Pat to reauthenticate:

```
clear uauth pat
```

# url-cache

Cache responses to URL filtering requests to the Websense server. (Configuration mode.)

**url-cache dst | src\_dst size**

**no url-cache dst | src\_dst size**

**clear url-cache**

**show url-cache stat**

## Syntax Description

<b>dst</b>	Cache entries based on the URL destination address. Select this mode if all users share the same URL filtering policy on the Websense server.
<b>src_dst</b>	Cache entries based on the both the source address initiating the URL request as well as the URL destination address. Select this mode if users do not share the same URL filtering policy on the Websense server.
<i>size</i>	Specify a value for the cache size within the range 1 to 128 KB.
<b>stat</b>	Use the <b>stat</b> option to display additional URL cache statistics, including the number of cache lookups and hit rate.

## Usage Guidelines

The **url-cache** command caches responses to URL filtering requests to the Websense server. Caching stores URL access privileges in memory on the PIX Firewall. When a host requests a connection, the PIX Firewall first looks in the URL cache for matching access privileges instead of forwarding the request to the Websense server. Disable caching with the **no url-cache** command. The **clear url-cache** command removes **url-cache** command statements from the configuration.



### Note

Access to the URL cache does not update the Websense accounting logs. Before using this command, let Websense run to accumulate logs to let you view Websense accounting information. After you get a usage profile that meets your security needs, enable this command to increase throughput.



### Note

If you change settings on the Websense server, disable the cache with the **no url-cache** command and then re-enable the cache with the **url-cache** command.

The **url-cache** command allows you to enable URL caching, set the size of the cache, and displays cache statistics.

The **show url-cache** command with the **stats** option displays the following entries:

- Size—The size of the cache in kilobytes, set with the **url-cache size** option.
- Entries—The maximum number of cache entries based on the cache size.
- In Use—The current number of entries in the cache.

- Lookups—The number of times the PIX Firewall has looked for a cache entry.
- Hits—The number of times the PIX Firewall has found an entry in the cache.

You can view additional information about Websense access with the **show perfmon** command.

### Examples

The following example caches all outbound HTTP connections based on the source and destination addresses:

```
url-cache src_dst 128
```

The following is sample output for the **show url-cache stat** command:

```
show url-cache stat
```

```
URL Filter Cache Stats
```

```
-----  
Size :      1KB  
Entries :    36  
In Use :    30  
Lookups :   300  
Hits :     290
```

# url-server

Designate a server running Websense for use with the **filter** command. (Configuration mode.)

```
url-server [(if_name)] host ip_address [timeout seconds] [protocol [TCP | UDP] version [1 | 4]
```

```
no url-server host ip_address
```

## Syntax Description

<i>if_name</i>	The network interface where the authentication server resides. If not specified, the default is inside.
<b>host</b> <i>ip_address</i>	The server that runs the Websense URL filtering application.
<b>timeout</b> <i>seconds</i>	The maximum idle time permitted before PIX Firewall switches to the next server you specified. The default is 5 seconds.
<b>protocol</b>	The protocol can be configured using <b>TCP</b> or <b>UDP</b> keywords. The default is TCP protocol, version 1.
<b>version</b>	The version of the protocol can be configured using <b>1</b> or <b>4</b> keywords. The default is TCP protocol, version 1. TCP can be configured using version 1 or version 4. UDP can be configured using version 4 only.

## Usage Guidelines

The **url-server** command designates a server that runs Websense, a URL filtering application. Once you designate the server, enable the URL filtering service with the **filter** command.




---

**Note** You can have a total of 16 URL servers.

---

Follow these steps to filter URLs:

- 
- Step 1** Designate a Websense server with the **url-server** command.
  - Step 2** Enable filtering with the **filter** command.
  - Step 3** If needed, improve throughput with the **url-cache** command. However, this command does not update Websense logs, which may affect Websense accounting reports. Accumulate Websense run logs before using the **url-cache** command.
  - Step 4** Use the **show url-cache stats** and the **show perfmon** commands to view run information.

Additional information on Websense is available at the following site:

<http://www.websense.com>

---

### Examples

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

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

# virtual

Access PIX Firewall virtual server. (Configuration mode.)

**virtual http** *ip\_address* [**warn**]

**virtual telnet** *ip\_address*

## Syntax Description

*ip\_address* For outbound use, *ip\_address* must be an address routed to the PIX Firewall. Use an RFC 1918 address that is not in use on any interface.

For inbound use, *ip\_address* must be an unused global address. An **access-list** and **static** command pair must provide access to *ip\_address*, as well as an **aaa authentication** command statement. See the “Examples” section for more information.

For example, if an inside client at 192.168.0.100 has a default gateway set to the inside interface of the PIX Firewall at 192.168.0.1, the *ip\_address* can be any IP address not in use on that segment (such as 10.2.3.4). As another example, if the inside client at 192.168.0.100 has a default gateway other than the PIX Firewall (such as a router at 192.168.0.254), then the *ip\_address* would need to be set to a value that would get statically routed to the PIX Firewall. This might be accomplished by using a value of 10.0.0.1 for the *ip\_address*, then on the client, setting the PIX Firewall at 192.168.0.1 as the route to host 10.0.0.1.

**warn** Let **virtual http** command users know that the command was redirected. This option is only applicable for text-based browsers where the redirect cannot happen automatically.

## Usage Guidelines

The **virtual http** command lets web browsers work correctly with the PIX Firewall **aaa** command. The **aaa** command assumes that the AAA server database is shared with a web server. PIX Firewall automatically provides the AAA server and web server with the same information. The **virtual http** command works with the **aaa** command to authenticate the user, separate the AAA server information from the web client’s URL request, and direct the web client to the web server. Use the **show virtual http** command to list commands in the configuration. Use the **no virtual http** command to disable its use.

The **virtual http** command works by redirecting the web browser’s initial connection to the *ip\_address*, which resides in the PIX Firewall, authenticating the user, then redirecting the browser back to the URL which the user originally requested. This mechanism comprises the PIX Firewall unit’s new virtual server feature. The reason this command is named as it is, is because the **virtual http** command accesses the virtual server for use with HTTP, another name for the Web. This command is especially useful for PIX Firewall interoperability with Microsoft IIS, but is useful for other authentication servers.

When using HTTP authentication to a site running Microsoft IIS that has “Basic text authentication” or “NT Challenge” enabled, users may be denied access from the Microsoft IIS server. This occurs because the browser appends the string: “Authorization: Basic=Uuhjksdkfhk==” to the HTTP GET commands. This string contains the PIX Firewall authentication credentials.

Windows NT Microsoft IIS servers respond to the credentials and assume that a Windows NT user is trying to access privileged pages on the server. Unless the PIX Firewall username password combination is exactly the same as a valid Windows NT username and password combination on the Microsoft IIS server, the HTTP GET command is denied.

To solve this problem, PIX Firewall provides the **virtual http** command which redirects the browser's initial connection to another IP address, authenticates the user, then redirects the browser back to the URL which the user originally requested.

Once authenticated, a user never has to reauthenticate no matter how low the PIX Firewall uauth timeout is set. This is because the browser caches the “Authorization: Basic=Uuhjksdkfhk==” string in every subsequent connection to that particular site. This can *only* be cleared when the user exits *all* instances of Netscape Navigator or Internet Explorer and restarts. Flushing the cache is of no use.




---

**Note** If you want double authentication through the authentication and web browser, configure the authentication server to not accept anonymous connections.

---




---

**Note** Do not set the **timeout uauth** duration to 0 seconds when using the **virtual** command because this will prevent HTTP connections to the real web server.

---




---

**Note** For both the **virtual http** and **virtual telnet** commands, if the connection is started on either an outside or perimeter interface, a **static** and **access-list** command pair is required for the fictitious IP address.

---

The **virtual telnet** command allows the Virtual Telnet server to provide a way to pre-authenticate users who require connections through the PIX Firewall using services or protocols that do not support authentication.

The **virtual telnet** command can be used both to log in and log out of the PIX Firewall. When an unauthenticated user Telnets to the virtual IP address, they are challenged for their username and password, and then authenticated with the TACACS+ or RADIUS server. Once authenticated, they see the message “Authentication Successful” and their authentication credentials are cached in the PIX Firewall for the duration of the uauth timeout.

If a user wishes to log out and clear their entry in the PIX Firewall uauth cache, the user can again Telnet to the virtual address. The user is prompted for their username and password, the PIX Firewall removes the associated credentials from the uauth cache, and the user will receive a “Logout Successful” message.

If inbound users on either the perimeter or outside interfaces need access to the Virtual Telnet server, a **static** and **access-list** command pair must accompany use of the **virtual telnet** command. The global IP address in the **static** command must be a real IP address. The local address in the **static** command is the IP address of the virtual server.

The Virtual Telnet server provides a way to pre-authenticate users who require connections through the PIX Firewall using services or protocols that do not support authentication. Users first connect to the Virtual Telnet server IP address, where the user is prompted for a username and password.

### Examples

- **virtual http**—The following example shows the commands required to use the **virtual http** command for an inbound connection:

```
static (inside, outside) 209.165.201.1 192.168.1.1 netmask 255.255.255.255
access-list acl_out permit tcp any host 209.165.201.1 eq 80
access-group acl_out in interface outside
aaa authentication include any inbound 192.168.1.1 255.255.255.255 0 0 tacacs+
virtual http 209.165.201.1
```

The next example displays the **show virtual** command output:

```
show virtual http
virtual http 209.165.201.1
```

- **virtual telnet**—After adding the **virtual telnet** command to the configuration and writing the configuration to Flash memory, users wanting to start PPTP sessions through PIX Firewall use Telnet to access the *ip\_address* as shown in the following example:

On the PIX Firewall:

```
virtual telnet 209.165.201.25
static (inside,outside) 209.165.201.25 10.8.8.11 netmask 255.255.255.255
access-list acl_out permit tcp any host 209.165.201.25 eq telnet
access-group acl_out in interface outside
write memory
```

On an inside host:

```
/unix/host%telnet 209.165.201.30
Trying 209.165.201.30...
Connected to 209.165.201.30.
Escape character is '^]'.

username: username

TACACS+ Password: password

Authentication Successful

Connection closed by foreign host.
/unix/host%
```

The *username* and *password* are those for the user on the TACACS+ server.

# vpng

Implements the PPTP feature. (Configuration mode.)

```

vpng enable if_name

vpng group group_name accept dialin pptp

vpng group group_name ppp authentication PAP | CHAP | MSCHAP

vpng group group_name ppp encryption mppe 40 | 128 | auto [required]

vpng group group_name client configuration address local address_pool_name

vpng group group_name client configuration dns dns_server_ip1 [dns_server_ip2]

vpng group group_name client configuration wins wins_server_ip1 [wins_server_ip2]

vpng group group_name client authentication aaa aaa_server_group

vpng group group_name client authentication local

vpng username username password password

vpng group group_name pptp echo echo_timeout

show vpng tunnel [id tunnel_id | packets | state | summary | transport]

show vpng username [username]

show vpng session [id session_id | username login_name | packets | state | window]

show vpng pppinterface [id intf_id]

clear vpng [group | username | tunnel [all | [id tunnel_id]]]

```

## Syntax Description

<b>enable</b> <i>if_name</i>	Enable the VPDN function on a PIX Firewall interface. Specify the interface in <i>if_name</i> where PPTP traffic is received. Only inbound connections are supported.
<b>group</b> <i>group_name</i>	Specify the VPDN group name. The VPDN <i>group_name</i> is an ASCII string to denote a VPDN group. You can make up the name. The maximum length of the name is 128 bytes.
<b>accept dialin pptp</b>	Accept a dial-in request using PPTP.

<b>ppp authentication PAP   CHAP   MSCHAP</b>	Specify the PPP (Point-to-Point Protocol) authentication protocol. The Windows client dial-up networking settings lets you specify what authentication protocol to use (PAP, CHAP, or MS-CHAP). Whatever you specify on the client must match the setting you use on the PIX Firewall. PAP (Password Authentication Protocol) lets PPP peers authenticate each other. PAP passes the host name or username in clear text. CHAP (Challenge Handshake Authentication Protocol) lets PPP peers prevent unauthorized access through interaction with an access server. MS-CHAP is a Microsoft derivation of CHAP. PIX Firewall supports MS-CHAP version 1 only (not version 2.0).
	If an authentication protocol is not specified on the host, do not specify the <b>ppp authentication</b> option in your configuration.
<b>ppp encryption mppe 40   128   auto [required]</b>	Specify the number of session key bits used for MPPE (Microsoft Point-to-Point Encryption) negotiation. The domestic version of the Windows client can support 40- and 128-bit session keys, but international version of the Windows client only supports 40-bit session keys. On the PIX Firewall, use <b>auto</b> to accommodate both. Use <b>required</b> to indicate that MPPE must be negotiated or the connection will be terminated.
<b>client configuration address local</b> <i>address_pool_name</i>	Specify the local address pool used to allocate an IP address to a client. Use the <b>ip local pool</b> command to specify the IP addresses for use by the clients.
<b>client configuration dns</b> <i>dns_server_ip1</i> <i>[dns_server_ip2]</i>	Specify up to two DNS server IP addresses. If set, the PIX Firewall sends this information to the Windows client during the IPCP phase of PPP negotiation.
<b>client configuration wins</b> <i>wins_server_ip1</i> <i>[wins_server_ip2]</i>	Specify up to two WINS server IP addresses.
<b>client authentication aaa</b> <i>aaa_server_group</i>	Specify the AAA server group for user authentication.
<b>client authentication local</b>	Authenticate using the local username and password entries you specify in the PIX Firewall configuration.
<b>password</b>	Specify local user password.
<b>pptp echo</b> <i>echo_timeout</i>	Specify the PPTP keep-alive echo timeout value in seconds. PIX Firewall terminates a tunnel if an echo reply is not received within the timeout period you specify.
<b>id</b> <i>tunnel_id</i>	Tunnel identification name.
<b>id</b> <i>session_id</i>	Session identification name.
<b>pppinterface id</b> <i>intf_id</i>	A PPP virtual interface is created for each PPTP tunnel. Use the <b>show vpdn session</b> command to display the interface identification value.

<b>username</b>	Enter or display local username.
<b>packets</b>	Packet and byte count.
<b>state</b>	Session state.
<b>summary</b>	Tunnel summary information.
<b>transport</b>	Tunnel transport information.
<b>window</b>	Window information.
<b>group</b>	[ <b>clear</b> command only]—Removes all <b>vpng group</b> commands from the configuration.
<b>username</b>	[ <b>clear</b> command only]—Removes all <b>vpng username</b> commands from the configuration.
<b>tunnel</b>	[ <b>clear</b> command only]—Removes one or more PPTP tunnels from the configuration.
<b>all</b>	[ <b>clear</b> command only]—Removes all PPTP tunnels from the configuration.
<b>id tunnel_id</b>	[ <b>clear</b> command only]—Removes PPTP tunnels from the configuration that match <i>tunnel_id</i> . You can view the tunnel IDs with the <b>show vpng tunnel</b> command.

### Usage Guidelines

The **vpng** command implements the PPTP feature for inbound connections between the PIX Firewall and a Windows client. Point-to-Point Tunneling Protocol (PPTP) is a layer 2 tunneling protocol, which lets a remote client use a public IP network to communicate securely with servers at a private corporate network. PPTP tunnels the IP protocol. RFC 2637 describes the PPTP protocol.

Only inbound PPTP connections are supported and only one PIX Firewall interface can have the **vpng** command enabled.

PPTP is an alternative to IPSec handling for VPN clients. While PPTP is less secure than IPSec, PPTP is easier to implement and maintain.

Supported authentication protocols include: PAP, CHAP, and MS-CHAP using external AAA (RADIUS or TACACS+) servers or the PIX Firewall local username and password database. Through the PPP IPCP protocol negotiation, PIX Firewall assigns a dynamic internal IP address to the PPTP client allocated from a locally defined IP address pool.

PIX Firewall PPTP VPN supports standard PPP CCP negotiations with Microsoft Point-To-Point Encryption (MPPE) extensions using RSA/RC4 algorithm. MPPE currently supports 40-bit and 128-bit session keys. MPPE generates an initial key during user authentication and refreshes the key regularly. In this release, compression is not supported.

When you specify MPPE, you must use the MS-CHAP PPP authentication protocol. If you are using an external AAA server, the protocol must be RADIUS and the external RADIUS server must be able to return the Microsoft MSCHAP\_MPPE\_KEY attribute to the PIX Firewall in the RADIUS Authentication Accept packet. See RFC-2548, "Microsoft Vendor Specific RADIUS Attributes," for more information on the MSCHAP\_MPPE\_KEY attribute.

Currently, Cisco has only tested the Steel-Belted RADIUS server from Funk Software as a server able to return the MSCHAP\_MPPE\_KEYS attribute.

PIX Firewall PPTP VPN has been tested with the following Microsoft Windows products: Windows 95 with DUN1.3, Windows 98, Windows NT 4.0 with Service Pack (SP) 6, and Windows 2000 Beta.

You can troubleshoot PPTP traffic with the **debug ppp** and **debug vpdn** commands.

Use the **vpdn** command with the **sysopt connection permit-pptp** to allow PPTP traffic to bypass checking of **conduit** or **access-list** command statements.

The **show vpdn** commands list tunnel and session information.

The **clear vpdn** command removes all **vpdn** commands from the configurations and stops all the active PPTP tunnels. The **clear vpdn all** command lets you remove all tunnels, and the **clear vpdn id tunnel\_id** command lets you remove tunnels associated with *tunnel\_id*. (You can view the *tunnel\_id* with the **show vpdn** command.) The **clear vpdn group** command removes all the **vpdn group** commands from the configuration. The **clear vpdn username** command removes all the **vpdn username** commands from the configuration. The **clear vpdn** command removes all **vpdn** commands from the configuration.

### Examples

The following examples list the output of the **show vpdn** commands.

The following is sample output for the **show vpdn tunnel** command:

```
show vpdn tunnel
PPTP Tunnel Information (Total tunnels=1 sessions=1)

Tunnel id 1, remote id is 1, 1 active sessions
  Tunnel state is estabd, time since event change 19 secs
  remote  Internet Address 209.165.201.1, port 1723
  Local   Internet Address 172.16.1.209, port 1723
  13 packets sent, 1269 received, 420 bytes sent, 120850 received
```

The following is sample output for the **show vpdn tunnel packet** command:

```
show vpdn tunnel packet
PPTP Tunnel Information (Total tunnels=1 sessions=1)

LocID  Pkts-In  Pkts-Out  Bytes-In  Bytes-Out
  1      1196      13        113910    420
```

The following is sample output for the **show vpdn tunnel state** command:

```
show vpdn tunnel state
PPTP Tunnel Information (Total tunnels=1 sessions=1)

LocID  RemID  State  Time-Since-Event-Chg
  1      1     estabd      6 secs
```

The following is sample output for the **show vpng tunnel summary** command:

```
show vpng tunnel summary
PPTP Tunnel Information (Total tunnels=1 sessions=1)

LocID RemID   State Remote Address  Port  Sessions
   1     1   estab 172.16.38.194 1723      1
```

The following is sample output for the **show vpng tunnel transport** command:

```
show vpng tunnel transport
PPTP Tunnel Information (Total tunnels=1 sessions=1)

LocID Type Local Address  Port Remote Address  Port
   1  IP 172.16.1.209 1723 172.16.38.194 1723
```

The following is sample output for the **show vpng session** command:

```
show vpng session
PPTP Session Information (Total tunnels=1 sessions=1)

Call id 1 is up on tunnel id 1
Remote Internet Address is 172.16.38.194
  Session username is aperson, state is estab
    Time since event change 6552 secs, interface outside
    Remote call id is 20484
    PPP interface id is 1
    13 packets sent, 1269 received, 420 bytes sent, 120850 received
    Seq 14, Ack 1268, Ack_Rcvd 13, peer RWS 64
    0 out of order packets
```

The following is sample output of a simple configuration that allows Windows PPTP clients to dial in without any authentication (not recommended). The Windows client can Telnet to internal host 192.168.0.2 through the static global address 209.165.201.2.

```
ip local pool my-addr-pool 10.1.1.1-10.1.1.254
vpng group 1 accept dialin pptp
vpng group 1 client configuration address local my-addr-pool
vpng enable outside
static (inside, outside) 209.165.201.2 192.168.0.2
access-list acl_out permit tcp 10.1.1.0 255.255.255.0 host 209.165.201.2 eq telnet
access-group acl_out in interface outside
```

In the next example, PPTP clients authenticate using MS-CHAP and negotiate MPPE encryption with the PIX Firewall. The PPTP client can Telnet to host 192.168.0.2 through the static global 209.165.201.2. The Telnet session will be encrypted.

```
ip local pool my-addr-pool 10.1.1.1-10.1.1.254
aaa-server my-aaa-server-group (inside) host 192.168.0.10 key 12345678
aaa-server my-aaa-server-group protocol radius
vpng group 1 accept dialin pptp
vpng group 1 ppp authentication mschap
vpng group 1 client authentication aaa my-aaa-server-group
vpng group 1 ppp encryption mppe auto required
vpng group 1 client configuration address local my-addr-pool
vpng enable outside
static (inside, outside) 209.165.201.2 192.168.0.2
access-list acl_out permit tcp 10.1.1.0 255.255.255.0 host 209.165.201.2 eq telnet
access-group acl_out in interface outside
```

In the next example, PPTP clients authenticate using MS-CHAP, negotiate MPPE encryption, receive the DNS and WINS server addresses, and can Telnet to the host 192.168.0.2 directly through the **nat 0** command statement.

```
ip local pool my-addr-pool 10.1.1.1-10.1.1.254
aaa-server my-aaa-server-group (inside) host 192.168.0.10 key 12345678
aaa-server my-aaa-server-group protocol radius
vpdn group 1 accept dialin pptp
vpdn group 1 ppp authentication mschap
vpdn group 1 ppp encryption mppe auto required
vpdn group 1 client configuration address local my-addr-pool
vpdn group 1 client authentication aaa my-aaa-server-group
vpdn group 1 client configuration dns 10.2.2.99
vpdn group 1 client configuration wins 10.2.2.100
vpdn enable outside
access-list nonat permit ip host 192.168.0.2 10.1.1.0 255.255.255.0
access-list nonat permit ip host 10.2.2.99 10.1.1.0 255.255.255.0
access-list nonat permit ip host 10.2.2.100 10.1.1.0 255.255.255.0
nat (inside) 0 access-list nonat
access-list acl_out permit tcp 10.1.1.0 255.255.255.0 host 192.168.0.2 eq telnet
access-list acl_out permit udp 10.1.1.0 255.255.255.0 host 10.2.2.99 eq domain
access-list acl_out permit udp 10.1.1.0 255.255.255.0 host 10.2.2.100 eq netbios-ns
access-group acl_out in interface outside
```

In the next example, PPTP clients authenticate using MS-CHAP, negotiate MPPE encryption, receive the DNS and WINS server addresses, and can Telnet to the host 192.168.0.2 directly through the **nat 0** command statement. An **access-group** command statement is not present because the **sysopt connection permit-pptp** command statement allows all the PPTP traffic through the tunnel.

```
ip local pool my-addr-pool 10.1.1.1-10.1.1.254
aaa-server my-aaa-server-group (inside) host 192.168.0.10 key 12345678
aaa-server my-aaa-server-group protocol radius
vpdn group 1 accept dialin pptp
vpdn group 1 ppp authentication mschap
vpdn group 1 ppp encryption mppe auto required
vpdn group 1 client configuration address local my-addr-pool
vpdn group 1 client authentication aaa my-aaa-server-group
vpdn group 1 client configuration dns 10.2.2.99
vpdn group 1 client configuration wins 10.2.2.100
vpdn enable outside
access-list nonat permit ip host 192.168.0.2 10.1.1.0 255.255.255.0
access-list nonat permit ip host 10.2.2.99 10.1.1.0 255.255.255.0
access-list nonat permit ip host 10.2.2.100 10.1.1.0 255.255.255.0
nat (inside) 0 access-list nonat
sysopt connection permit-pptp
```

In the next example, PPTP clients authenticate using MS-CHAP, negotiate MPPE encryption, receive the DNS and WINS server addresses, and can Telnet to the host 192.168.0.2 directly through the **nat 0** command. The PPTP authenticates using the PIX Firewall local username and password database you create with the **vpng username** command. Users are reauthenticated again by the **aaa** command when they start a Telnet session. An **access-group** command statement is not present because the **sysopt connection permit-pptp** command statement allows all the PPTP traffic through the tunnel.

```
ip local pool my-addr-pool 10.1.1.1-10.1.1.254
aaa-server my-aaa-server-group (inside) host 192.168.0.10 key 12345678
aaa-server my-aaa-server-group protocol radius
vpng username usrname1 password password1
vpng group 1 accept dialin pptp
vpng group 1 ppp authentication mschap
vpng group 1 ppp encryption mppe auto required
vpng group 1 client configuration address local my-addr-pool
vpng group 1 client authentication local
vpng group 1 client configuration dns 10.2.2.99
vpng group 1 client configuration wins 10.2.2.100
vpng enable outside
access-list nonat permit ip host 192.168.0.2 10.1.1.0 255.255.255.0
access-list nonat permit ip host 10.2.2.99 10.1.1.0 255.255.255.0
access-list nonat permit ip host 10.2.2.100 10.1.1.0 255.255.255.0
nat (inside) 0 access-list nonat
sysopt connection permit-pptp
aaa authentication include telnet inbound 192.168.0.2 255.255.255.255 10.1.1.0
255.255.255.0
```

# who

Show active Telnet administration sessions on the PIX Firewall. (Unprivileged mode.)

```
who [local_ip]
```

```
show who [local_ip]
```

## Syntax Description

*local\_ip* An optional internal IP address to limit the listing to one IP address or to a network IP address.

## Usage Guidelines

The **who** command shows the PIX Firewall TTY\_ID and IP address of each Telnet client currently logged into the PIX Firewall. This command is the same as the **show who** command.

See also: **kill**, **telnet**.

## Examples

The following example shows how to display the current Telnet sessions:

```
who  
2: From 192.168.2.2  
1: From 192.168.1.3
```

# write

Store, view, or erase the current configuration. (Privileged mode.)



## Note

The PIX 506 does not support use of the **write standby** command. Also, the PIX 515, PIX 506, and the PIX 525 do not support use of the **write floppy** command.

**write net** [[*server\_ip*]:[*filename*]]

**write erase**

**write floppy**

**write memory**

**write standby**

**write terminal**

## Syntax Description

*server\_ip* Store current configuration at a host available across the network. If you specify the full path and filename in the **tftp-server** command, only specify a colon (:) in the **write** command.

*filename* A filename you specify to qualify the location of the configuration file on the TFTP server named in *server\_ip*. If you set a filename with the **tftp-server** command, do not specify it in the **write** command; instead just use a colon (:) without a filename.

Many TFTP servers require the configuration file to be world-writable to write to it.

**erase** Clear the Flash memory configuration.

**floppy** Store current configuration on diskette.

**memory** Store current configuration in Flash memory.

**standby** Store configuration to the failover Standby unit from RAM to RAM.

**terminal** Display current configuration on the terminal.

## Usage Guidelines

The **write net** command stores the current configuration into a file on a TFTP server elsewhere in the network. Additionally, the **write net** command uses the TFTP server IP address specified in the **tftp-server** command.

If you specify both the IP address and path name in the **tftp-server** command, you can specify the **write net :filename** as simply a colon (:). For example:

```
write net :
```

Use the **configure net** command to get the configuration from the file.

The **write erase** command clears the Flash memory configuration.

The **write floppy** command stores the current configuration on diskette. The diskette must be DOS formatted or a PIX Firewall boot disk. If you are formatting the diskette from Windows, choose the Full format type, not the Quick (erase) selection. You can tell that information is stored on the diskette by observing that the light next to the diskette drive glows while information transfers.

The diskette you create can only be read or written by the PIX Firewall. If you use the **write floppy** command with a diskette that is not a PIX Firewall boot disk, do not leave the floppy in the floppy drive because it will prevent the firewall from rebooting in the event of a power failure or system reload. Only one copy of the configuration can be stored on a single diskette.

The **write memory** command saves the current running configuration to Flash memory. Use the **configure memory** command to merge the current configuration with the image you saved in Flash memory.

PIX Firewall lets processing continue during the **write memory** command.

If another PIX Firewall console user tries to change the configuration while you are executing the **write memory** command, the user receives the following messages:

```
Another session is busy writing configuration to memory  
Please wait a moment for it to finish
```

After the **write memory** command completes, PIX Firewall lets the other command complete.

**Note**

---

Only use the **write memory** command if a configuration has been created with IP addresses for both network interfaces.

---

The **write standby** command writes the configuration stored in RAM on the Active failover unit to the RAM on the Standby unit. When the Primary unit boots it automatically writes the configuration to the Secondary unit. Use the **write standby** command if the primary and secondary units' configurations have different information.

The **write terminal** command displays the current configuration in the PIX Firewall unit's RAM memory.

You can also display the configuration stored in Flash memory using the **show configure** command.

See also: **configure**.

### Examples

The following example specifies a configuration file on the TFTP server and then stores the configuration in the new\_config file:

```
tftp-server 10.1.1.2 /pixfirewall/config/new_config  
write net :
```

The following example erases the contents of Flash memory and reloads the PIX Firewall:

```
write erase  
Erase PIX configuration in Flash memory? [confirm] y  
reload
```

The following example saves the configuration on diskette:

```
write floppy  
Building configuration..  
[OK]
```

The following example saves the current configuration to Flash memory:

```
write memory  
Building configuration..  
[OK]
```

The following example displays the configuration:

```
write terminal  
Building configuration..  
: Saved  
...
```

# xlate (clear and show)

View or clear translation slot information. (Privileged mode.)

```
clear xlate [global | local ip1[-ip2] [netmask mask]] lport | gport port[-port]]
[interface if1[,if2][,ifn]] [state static [,dump] [,portmap] [,norandomseq] [,identity]]
```

```
show xlate [global | local ip1[-ip2] [netmask mask]] lport | gport port[-port]]
[interface if1[,if2][,ifn]] [state static [,dump] [,portmap] [,norandomseq] [,identity]]
```

## Syntax Description

<b>[global   local</b> <i>ip1</i> [- <i>ip2</i> ] <b>[netmask</b> <i>mask</i> ]	Display active translations by global IP address or local IP address using the network mask to qualify the IP addresses.
<b>lport</b>   <b>gport</b> <i>port</i> [- <i>port</i> ]	Display active translations by local and global port specifications. See “Ports” in Chapter 1, “Introduction” for a list of valid port literal names.
<b>interface</b> <i>if1</i> [, <i>if2</i> ][,ifn]	Display active translations by interface.
<b>state</b>	Display active translations by state; <b>static</b> translation ( <b>static</b> ), <b>dump</b> (cleanup), PAT <b>global</b> ( <b>portmap</b> ), a <b>nat</b> or <b>static</b> translation with the <b>norandomseq</b> setting ( <b>norandomseq</b> ), or the use of the <b>nat 0</b> , identity feature ( <b>identity</b> ).

## Usage Guidelines

The **clear xlate** command clears the contents of the translation slots. (“xlate” means translation slot.) The **show xlate** command displays the contents of only the translation slots.

Translation slots can persist after key changes have been made. Always use the **clear xlate** command after adding, changing, or removing the **aaa-server**, **access-list**, **alias**, **conduit**, **global**, **nat**, **route**, or **static** commands in your configuration.

See also: **show conn**, **timeout**, **uauth**.

## Examples

The following is sample output for two static translations, the first with two associated connections (called “nconns”) and the second with four.

```
show xlate
Global 209.165.201.10 Local 209.165.201.10 static nconns 1 econns 0
Global 209.165.201.30 Local 209.165.201.30 static nconns 4 econns 0
```

■ xlate (clear and show)