



## S Commands

### service

Reset inbound connections. (Configuration mode.)

| Configure with the command...     | Remove with the command...           |
|-----------------------------------|--------------------------------------|
| <code>service resetinbound</code> | <code>no service resetinbound</code> |
| <code>service resetoutside</code> | <code>clear service</code>           |

| Show command options      | Show command output                             |
|---------------------------|---|
| <code>show service</code> | Displays service commands in the configuration. |

#### Syntax Description

|                           |   |
|---------------------------|---|
| <code>resetinbound</code> | Reset inbound connections.                  |
| <code>resetoutside</code> | Reset connections on the outside interface. |

#### 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
```

If you use the **resetoutside** command, the PIX Firewall actively resets denied TCP packets that terminate at the PIX Firewall unit's least-secure interface. By default, these packets are silently discarded. The **resetoutside** option is highly recommended with dynamic or static interface Port Address Translation (PAT). The static interface PAT is available with PIX Firewall version 6.0 and higher. This option allows the PIX Firewall to quickly terminate the identity request (IDENT) from an external SMTP or FTP server. Actively resetting these connections avoids the thirty-second time-out delay.

If you wish to remove **service** command statements from the configuration, use the **clear service** command.

# setup

The **setup** command prompts you to enter the information needed to use the Cisco PIX Device Manager (PDM) with a new PIX Firewall. (Configuration Mode.)

| Start configuration with the command... | Remove with the command... |
|---|----------------------------|
| <b>setup</b>                            | N/A                        |

## Syntax Description

|              |  |
|--------------|--|
| <b>setup</b> | Asks for the information needed to start using a new PIX Firewall unit if no configuration is found in the Flash memory. |
|--------------|--|

## Usage Guidelines

The PIX Firewall requires some pre-configuration before PDM can connect to it. (The setup dialog automatically appears at boot time if there is no configuration in the Flash memory. Once you enter the **setup** command, you will be asked for the setup information in [Table 8-1](#).)

**Table 8-1 PIX Firewall Setup Information**

| Prompt                       | Description  |
|------------------------------|--|
| Enable password:             | Specify an enable password for this PIX Firewall. (The password must be at least three characters long.) |
| Clock (UTC)                  | Set the PIX Firewall clock to Universal Coordinated Time (also known as Greenwich Mean Time).            |
| Year [ <i>system year</i> ]: | Specify current year, or default to the year stored in the host computer.                                |

**Table 8-1 PIX Firewall Setup Information (continued)**

|  |   |
|--|---|
| Month [ <i>system month</i> ]:                 | Specify current month, or default to the month stored in the host computer.   |
| Day [ <i>system day</i> ]:                     | Specify current day, or default to the day stored in the host computer.   |
| Time [ <i>system time</i> ]                    | Specify current time in <i>hh:mm:ss</i> format, or default to the time stored in the host computer.   |
| Inside IP address:                             | Network interface IP address of the PIX Firewall.   |
| Inside network mask:                           | A network mask that applies to the inside IP address must be a valid mask such as 255.0.0.0, 255.255.0.0, or 255.255.x.x, etc. Use <b>0.0.0.0</b> to specify a default route. The <b>0.0.0.0</b> netmask can be abbreviated as <b>0</b> .   |
| Host name:                                     | The host name you want to display in the PIX Firewall command line prompt.  |
| Domain name:                                   | The DNS domain name of the network on which the PIX Firewall runs, for example <i>example.com</i> .   |
| IP address of host running PIX Device Manager: | IP address on which PDM connects to the PIX Firewall.   |
| Use this configuration and write to flash?     | Store the new configuration to Flash memory. Same as the <b>write memory</b> command. If the answer is <b>yes</b> , the inside interface will be enabled and the requested configuration will be written to Flash memory. If the user answers anything else, the setup dialog repeats using the values already entered as the defaults for the questions. |

The host and domain names are used to generate the default certificate for the SSL connection. The interface type is determined by the hardware.

### Examples

The following example shows how to complete the **setup** command prompts.

```
router (config)# setup
Pre-configure PIX Firewall now through interactive prompts [yes]? y
Enable Password [<use current password>]: ciscopix
Clock (UTC)
  Year [2001]: 2001
  Month [Aug]: Sep
  Day [27]: 12
  Time [22:47:37]: <Enter>
Inside IP address: 192.168.1.1
Inside network mask: 255.255.255.0
Host name: accounting_pix
Domain name: example.com
IP address of host running PIX Device Manager: 192.168.1.2
```

The following configuration will be used:

```
Enable Password: ciscopix
Clock (UTC): 22:47:37 Sep 12 2001
Inside IP address: ...192.168.1.1
Inside network mask: ...255.255.255.0
Host name: ...accounting_pix
Domain name: ...example.com
IP address of host running PIX Device Manager: ...192.168.1.2
```

Use this configuration and write to flash? y

**Related Commands**

- [aaa authentication](#)
- [ca](#)
- [copy](#)
- [http](#)

## session

Access an embedded AccessPro router console; only use this command if you have an AccessPro router installed in your PIX Firewall. (Privileged mode.)

| Start with the command...   | End with the command... |
|-----------------------------|-------------------------|
| <code>session enable</code> | <code>no session</code> |

| Show command options      | Show command output |
|---------------------------|---------------------|
| <code>show session</code> | Displays sessions.  |

**Note**

The PIX 506/506E and PIX 515/515E do not support use of the **session** command.

**Syntax Description**

|               |   |
|---------------|---|
| <b>enable</b> | Enable the <b>session</b> 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**

The following 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. (All modes.)

| Show command options       | Show command output   |
|----------------------------|---|
| <b>show</b> <i>command</i> | Runs the show command option specified. See individual commands for their show options. |
| <b>show</b> ?              | Displays a list of all commands available on the PIX Firewall.                          |

**Syntax Description**

|                |   |
|----------------|---|
| <i>command</i> | Any argument or list of arguments that specifies the information to display. Most commands have a <b>show</b> command form where the command name is used as <b>show</b> argument. For example, the <b>global</b> command has an associated <b>show global</b> command. |
|----------------|---|

**Usage Guidelines**

Explanations for the **show** form of specific commands are with the command. For example, the **show arp** command description is included with the **arp** command.

**Examples**

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

```

pixfirewall(config)# show ?
aaa                Enable, disable, or view TACACS+, RADIUS or LOCAL
                   user authentication, authorization and accounting
aaa-server         Define AAA Server group
access-group       Bind an access-list to an interface to filter inbound traffic
access-list        Add an access list
activation-key     Modify activation-key.
age                This command is deprecated. See ipsec, isakmp, map, ca commands
alias              Administer overlapping addresses with dual NAT.
apply              Apply outbound lists to source or destination IP addresses
arp                Change or view the arp table, and set the arp timeout value
auth-prompt        Customize authentication challenge, reject or acceptance prompt
blocks            Show system buffer utilization
ca                 CEP (Certificate Enrollment Protocol)
capture            Create and enroll RSA key pairs into a PKI (Public Key Infrastr.
capture            Capture inbound and outbound packets on one or more interfaces

```

|                   |  |
|-------------------|--|
| checksum          | View configuration information cryptochecksum  |
| chunkstat         | Display chunk stats  |
| clock             | Show and set the date and time of PIX  |
| conduit           | Add conduit access to higher security level network or ICMP  |
| configure         | Configure from terminal, floppy, memory, network, or factory-default. The configuration will be merged with the active configuration except for factory-default in which case the active configuration is cleared first. |
| conn              | Display connection information   |
| cpu               | Display cpu usage  |
| crypto            | Configure IPsec, IKE, and CA   |
| curpriv           | Display current privilege level  |
| debug             | Debug packets or ICMP tracings through the PIX Firewall.   |
| dhcpd             | Configure DHCP Server  |
| domain-name       | Change domain name   |
| dynamic-map       | Specify a dynamic crypto map template  |
| eeprom            | show or reprogram the 525 onboard i82559 devices   |
| enable            | Configure enable passwords   |
| established       | Allow inbound connections based on established connections   |
| failover          | Enable/disable PIX failover feature to a standby PIX   |
| filter            | Enable, disable, or view URL, Java, and ActiveX filtering  |
| fixup             | Add or delete PIX service and feature defaults   |
| flashfs           | Show, destroy, or preserve filesystem information  |
| fragment          | Configure the IP fragment database   |
| global            | Specify, delete or view global address pools, or designate a PAT(Port Address Translated) address  |
| h225              | Show the current h225 data stored for each connection.   |
| h245              | List the h245 connections.   |
| h323-ras          | Show the current h323 ras data stored for each connection.   |
| history           | Display the session command history  |
| http              | Configure HTTP server  |
| icmp              | Configure access for ICMP traffic that terminates at an interface  |
| interface         | Identify network interface type, speed duplex, and if shutdown   |
| igmp              | Clear or display IGMP groups   |
| ip                | Set the ip address and mask for an interface<br>Define a local address pool<br>Configure Unicast RPF on an interface<br>Configure the Intrusion Detection System   |
| ipsec             | Configure IPSEC policy   |
| isakmp            | Configure ISAKMP policy  |
| local-host        | Display or clear the local host network information  |
| logging           | Enable logging facility  |
| map               | Configure IPsec crypto map   |
| memory            | System memory utilization  |
| mroute            | Configure a multicast route  |
| mtu               | Specify MTU(Maximum Transmission Unit) for an interface  |
| multicast         | Configure multicast on an interface  |
| name              | Associate a name with an IP address  |
| nameif            | Assign a name to an interface  |
| names             | Enable, disable or display IP address to name conversion   |
| nat               | Associate a network with a pool of global IP addresses   |
| object-group      | Create an object group for use in 'access-list', 'conduit', etc  |
| ntp               | Configure Network Time Protocol  |
| outbound          | Create an outbound access list   |
| pager             | Control page length for pagination   |
| passwd            | Change Telnet console access password  |
| pdm               | Configure Pix Device Manager   |
| privilege         | Configure/Display privilege levels for commands  |
| processes         | Display processes  |
| remote-management | Configure remote management support  |
| rip               | Broadcast default route or passive RIP   |
| route             | Enter a static route for an interface  |
| username          | Configure user authentication local database   |
| service           | Enable system services   |

|              |  |
|--------------|--|
| session      | Access an internal AccessPro router console  |
| shun         | Manages the filtering of packets from undesired hosts  |
| snmp-server  | Provide SNMP and event information   |
| split-dns    | Configure split DNS resolution.  |
| ssh          | Add SSH access to PIX console, set idle timeout, display list of active SSH sessions & terminate a SSH session |
| static       | Configure one-to-one address translation rule  |
| sysopt       | Set system functional option   |
| tech-support | Tech support   |
| telnet       | Add telnet access to PIX console and set idle timeout  |
| terminal     | Set terminal line parameters   |
| tftp-server  | Specify default TFTP server address and directory  |
| timeout      | Set the maximum idle times   |
| traffic      | Counters for traffic statistics  |
| uauth        | Display or clear current user authorization information  |
| url-cache    | Enable URL caching   |
| url-block    | Enable URL pending block buffer and long URL support   |
| url-server   | Specify a URL filter server  |
| version      | Display PIX system software version  |
| virtual      | Set address for authentication virtual servers   |
| vpdn         | Configure VPDN (PPTP, L2TP, PPPoE) Policy  |
| vpnclient    | Configure VPN Client   |
| vpngroup     | Configure a policy group for VPN clients   |
| who          | Show active administration sessions on PIX   |
| xlate        | Display current translation and connection slot information  |

## show blocks/clear blocks

Show system buffer utilization. (Privileged mode.)

| Display with the command... | Clear buffers with the command... |
|-----------------------------|-----------------------------------|
| show blocks                 | clear blocks                      |

### Syntax Description

| blocks | The blocks in the preallocated system buffer. |
|--------|---|
|--------|---|

### Usage Guidelines

The **show blocks** command lists preallocated system buffer utilization. In the **show blocks** command 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.

**Examples**

The following is sample output from 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 command options | Show command output   |
|----------------------|---|
| <b>show checksum</b> | Displays four groups of hexadecimal numbers that act as a digital summary of the contents of the configuration. |

**Syntax Description**

|                 |   |
|-----------------|---|
| <b>checksum</b> | The hexadecimal numbers that act as a digital summary of the contents of the configuration. |
|-----------------|---|

**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 from the **show checksum** command:

```
show checksum
Cryptochecksum: 1a2833c0 129ac70b 1a88df85 650dbb81
```

# show conn

Display all active connections. (Privileged mode.)

| Show command options   | Show command output  |
|--|--|
| <b>show conn</b> [ <b>detail</b> ] [ <b>count</b> ] [ <b>foreign</b>   <b>local</b><br><i>ip</i> [- <i>ip2</i> ]] [ <b>netmask</b> <i>mask</i> ] [ <b>protocol</b> <b>tcp</b><br>  <b>udp</b>   <i>protocol</i> ] [ <b>fport</b>   <b>lport</b> <i>port1</i><br>[- <i>port2</i> ]] [ <b>state</b> [ <b>up</b> [, <b>finin</b> ] [, <b>finout</b> ]<br>[, <b>http_get</b> ] [, <b>sip</b> ] [, <b>smtp_data</b> ]<br>[, <b>smtp_banner</b> ] [, <b>smtp_incomplete</b> ]<br>[, <b>nojava</b> ] [, <b>data_in</b> ] [, <b>data_out</b> ]<br>[, <b>sqlnet_fixup_data</b> ] [, <b>conn_inbound</b> ]<br>[, <b>rpc</b> ] [, <b>h323</b> ] [, <b>dump</b> ]] | Displays the number of, and information about, the active connections for the options specified. |
| <b>show conn</b> [ <b>detail</b> ] [ <b>count</b> ]   [ <b>protocol</b><br><b>tcpludp</b> ] [ <b>foreignlocal</b> <i>ip1</i> [- <i>ip2</i> ]<br>[ <b>netmask</b> <i>mask</i> ]] [ <b>lportlfport</b><br><i>port1</i> [- <i>port2</i> ]] [ <b>state</b> <b>up</b> [, <b>finin</b> ] [, <b>finout</b> ]<br>[, <b>http_get</b> ] [, <b>smtp_data</b> ] [, <b>nojava</b> ]<br>[, <b>data_in</b> ] [, <b>data_out</b> ] [, <b>rpc</b> ] [, <b>h323</b> ]<br>[, <b>sqlnet_fixup_data</b> ] [, <b>conn_inbound</b> ]<br>[, <b>sip</b> ]]  | Displays the number of, and information about, the active connections for the options specified. |
| <b>show conn state</b> [ <b>detail</b> ] <b>up</b> [, <b>finin</b> ] [, <b>finout</b> ]<br>[, <b>http_get</b> ] [, <b>smtp_data</b> ]<br>[, <b>nojava</b> ] [, <b>data_in</b> ] [, <b>data_out</b> ] [, <b>rpc</b> ]<br>[, <b>h323</b> ] [, <b>sqlnet_fixup_data</b> ]<br>[, <b>conn_inbound</b> ] [, <b>sip</b> ] ]   | Displays the number of, and information about, the active connections for the options specified. |

## Syntax Description

|   |  |
|---|--|
| <b>count</b>  | Display only the number of used connections. The precision of the displayed count may vary depending on traffic volume and the type of traffic passing through the PIX Firewall unit.  |
| <b>detail</b>   | If specified, displays translation type and interface information.   |
| <b>foreign</b>   <b>local</b> <i>ip</i> [- <i>ip2</i> ]<br><b>netmask</b> <i>mask</i> | Display active connections by the foreign IP address or by local IP address. Qualify foreign or local active connections by network mask.  |
| <b>fport</b>   <b>lport</b> <i>port1</i><br>[- <i>port2</i> ]                         | Display foreign or local active connections by port. See “Ports” in Chapter 2, “Using PIX Firewall Commands” for a list of valid port literal names.   |
| <b>protocol</b> <b>tcp</b>   <b>udp</b>  <br><i>protocol</i>                          | Display active connections by protocol type. <i>protocol</i> is a protocol specified by number. See “Protocols” in Chapter 2, “Using PIX Firewall Commands” for a list of valid protocol literal names.  |
| <b>state</b>  | Display active connections by their current state: up ( <b>up</b> ), FIN inbound ( <b>finin</b> ), FIN outbound ( <b>finout</b> ), HTTP get ( <b>http_get</b> ), SMTP mail data ( <b>smtp_data</b> ), SIP connection ( <b>sip</b> ), SMTP mail banner ( <b>smtp_banner</b> ), incomplete SMTP mail connection ( <b>smtp_incomplete</b> ), an outbound command denying access to Java applets ( <b>nojava</b> ), inbound data ( <b>data_in</b> ), outbound data ( <b>data_out</b> ), SQL*Net data fix up ( <b>sqlnet_fixup_data</b> ), inbound connection ( <b>conn_inbound</b> ), RPC connection ( <b>rpc</b> ), H.323 connection ( <b>h323</b> ), dump clean up connection ( <b>dump</b> ). |

**Usage Guidelines**

The **show conn** command displays the number of, and information about, active TCP connections. You can also view the connection count information from the **show conn** command using SNMP.

The **show conn detail** command displays the following information:

```
{UDP | TCP} outside_ifc:real_addr/real-port [(map_addr/port)] inside_ifc:real_addr/real_port
[(map-addr/port)] flags flags
```

The connection flags are defined in [Table 8-2](#).

**Table 8-2 Connection Flags**

| Flag | Description                 |
|------|-----------------------------|
| U    | up                          |
| f    | inside FIN                  |
| F    | outside FIN                 |
| r    | inside acknowledged FIN     |
| R    | outside acknowledged FIN    |
| s    | awaiting outside SYN        |
| S    | awaiting inside SYN         |
| M    | SMTP data                   |
| H    | HTTP get (not used)         |
| T    | TCP SIP connection          |
| ---  | SKINNY (not used)           |
| I    | inbound data                |
| O    | outbound data               |
| q    | SQL*Net data                |
| d    | dump                        |
| P    | inside back connection      |
| E    | outside back connection     |
| G    | group                       |
| p    | replicated (unused)         |
| a    | awaiting outside ACK to SYN |
| A    | awaiting inside ACK to SYN  |
| B    | initial SYN from outside    |
| R    | RPC                         |
| H    | H.323                       |
| T    | UDP SIP connection          |
| m    | SIP media connection        |
| t    | SIP transient connection    |
| D    | DNS                         |

**Examples**

The following example shows a TCP session connection from inside host 10.1.1.15 to the outside telnet server at 192.150.49.10. Because there is no B flag, the connection is initiated from the inside. The "U", "I", and "O" flags denote that the connection is active and has received inbound and outbound data.

```
pixfirewall(config)# show conn
2 in use, 2 most used
TCP out 192.150.49.10:23 in 10.1.1.15:1026 idle 0:00:22
Bytes 1774 flags UIO
UDP out 192.150.49.10:31649 in 10.1.1.15:1028 idle 0:00:14
flags D-
```

The following example shows a UDP connection from outside host 192.150.49.10 to inside host 10.1.1.15. The D flag denotes that this is a DNS connection. The number 1028 is the DNS ID over the connection.

```
pixfirewall(config)# show conn detail
2 in use, 2 most used
Flags: A - awaiting inside ACK to SYN, a - awaiting outside ACK to SYN,
       B - initial SYN from outside, D - DNS, d - dump,
       E - outside back connection, f - inside FIN, F - outside FIN,
       G - group, H - H.323, I - inbound data, M - SMTP data,
       O - outbound data, P - inside back connection,
       Q - SQL*Net data, R - outside acknowledged FIN,
       R - UDP RPC, r - inside acknowledged FIN, S - awaiting inside SYN,
       s - awaiting outside SYN, U - up
TCP outside:192.150.49.10/23 inside:10.1.1.15/1026 flags UIO
UDP outside:192.150.49.10/31649 inside:10.1.1.15/1028 flags dD
```

The following is sample output from 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 website at 209.165.201.1. The global address on the outside interface is 209.165.201.7.

## show cpu usage

The **show cpu usage** command displays CPU utilization. (Privileged or configuration mode.)

| Show command options  | Show command output   |
|-----------------------|---|
| <b>show cpu usage</b> | Displays central processing unit (CPU) utilization information. |

**Syntax Description**

|                  |   |
|------------------|---|
| <b>cpu usage</b> | The central processing unit (CPU) usage data. |
|------------------|---|

**Usage Guidelines**

The **show cpu usage** command displays the central processing unit (CPU) usage information.

**Examples**

The following example shows the **show cpu usage** command output:

```
CPU utilization for 5 seconds: p1%; 1 minute: p2%; 5 minutes: p3%
```

The percentage usage prints as NA (not applicable) if the usage is unavailable for the specified time interval. This can happen if the user asks for CPU usage before the 5-second, 1-minute, or 5-minute time interval has elapsed.

## show crypto engine

Shows cryptography engine statistics.

```
show crypto engine
```

**Syntax Description**

|                      |   |
|----------------------|---|
| <b>crypto engine</b> | Displays usage statistics for the firewall cryptography engine. |
|----------------------|---|

**Command Modes**

Privileged or configuration mode.

**Usage Guidelines**

The **show crypto engine** command displays usage statistics for the cryptography engine used by the firewall.

**Examples**

The following example shows sample output for the **show crypto engine** command:

```
pixfirewall# show crypto engine
Crypto Engine Connection Map:
  size = 8, free = 6, used = 1, active = 1
```

In this command output, *size* is total number of unidirectional IPSec tunnels, *free* is the number of unused unidirectional IPSec tunnels, *used* is the number of allocated unidirectional IPSec tunnels, and *active* is the number of active unidirectional IPSec tunnels. Because tunnel 0 is reserved for system use, *size* is equal to *free* plus *used* plus one.

## show history

Display previously entered commands. (Privileged mode.)

| Show command options | Show command output                       |
|----------------------|---|
| <b>show history</b>  | Displays the previously entered commands. |

**Syntax Description**

|                |                               |
|----------------|-------------------------------|
| <b>history</b> | The list of previous entries. |
|----------------|-------------------------------|

**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 from the **show history** command:

```
show history
  enable
  ...
```

## show local-host/clear local host

View local host network states. (Privileged mode (**show**), configuration mode (**clear**).)

| Display with the command... | Clear with the command...                     |
|-----------------------------|---|
| <b>show local-host</b>      | <b>clear local-host</b> [ <i>ip_address</i> ] |

  

| Show command options                         | Show command output   |
|--|---|
| <b>show local-host</b> [ <i>ip_address</i> ] | Displays the network states of local hosts, and the number of hosts that are counted toward license limits if applicable. |

**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.

The **show local-host detail** command displays more information about active xlates and connections.

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.

On a PIX 501, cleared hosts are released from the license limit. You can view the number of hosts that are counted toward the license limit with the **show local-host** command.

**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 following is sample command output from the **show local-host** command:

```

pixfirewall(config)# show local-host
local host: <10.1.1.15>, conn(s)/limit = 2/0, embryonic(s)/limit = 0/0
  Xlate(s):
    PAT Global 192.150.49.1(1024) Local 10.1.1.15(516)
    PAT Global 192.150.49.1(0) Local 10.1.1.15 ICMP id 340
    PAT Global 192.150.49.1(1024) Local 10.1.1.15(1028)
  Conn(s):
    TCP out 192.150.49.10:23 in 10.1.1.15:1026 idle 0:00:25
      Bytes 1774 flags UIO
    UDP out 192.150.49.10:31649 in 10.1.1.15:1028 idle 0:00:17
      flags D-

```

For comparison, the following is sample command output from the **show local-host detail** command:

```

pixfirewall(config)# show local-host detail
local host: <10.1.1.15>, conn(s)/limit = 2/0, embryonic(s)/limit = 0/0
  Xlate(s):
    TCP PAT from inside:10.1.1.15/1026 to outside:192.150.49.1/1024
      flags ri
    ICMP PAT from inside:10.1.1.15/21505 to outside:192.150.49.1/0
      flags ri
    UDP PAT from inside:10.1.1.15/1028 to outside:192.150.49.1/1024
      flags ri
  Conn(s):
    TCP outside:192.150.49.10/23 inside:10.1.1.15/1026 flags UIO
    UDP outside:192.150.49.10/31649 inside:10.1.1.15/1028 flags dD

```

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.

# show memory

Show system memory utilization. (Privileged mode.)

| Show command options | Show command output                             |
|----------------------|---|
| <b>show memory</b>   | Displays system memory utilization information. |

## Syntax Description

**memory** The system memory data.

## 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.

## Examples

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

```
show memory
nnnnnnnn bytes total, nnnnnnnn bytes free
```

# show processes

Display processes. (Privileged mode.)

| Show command options  | Show command output                       |
|-----------------------|---|
| <b>show processes</b> | Displays a list of the running processes. |

## Syntax Description

**processes** The processes running on the PIX Firewall.

## Usage Guidelines

The **show processes** command displays a list of the 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 from the **show processes** command:

```
show processes
  PC      SP      STATE      Runtime      SBASE      Stack Process
Lsi 800125de 803603d0 80075ba0          0 8035f410 4004/4096 arp_timer
...
```

# show running-config

Display the PIX Firewall running configuration. (Privileged mode.)

| Show command options       | Show command output   |
|----------------------------|---|
| <b>show running-config</b> | Displays the configuration currently running on the PIX Firewall. |

## Syntax Description

**running-config** The configuration running on the PIX Firewall.

## Usage Guidelines

The **show running-config** command displays the current running configuration. The keyword **running-config** is used to match Cisco IOS software command. The **show running-config** command output is the same as the pre-existing PIX Firewall **write terminal** command.

The **running-config** keyword can be used only in the **show running-config** command. It cannot be used with **no** or **clear**, or as a standalone command. If it is, the CLI treats it as a non-supported command. Also, for this reason, when **?**, **no ?**, or **clear ?** are entered, a **running-config** option is not listed in the command list.

## Examples

The following is sample output from the **show running-config** command:

```

pixfirewall# show running-config
: Saved
:
PIX Version 6.2(1)
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 intf2 security10
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname pixdoc515
domain-name cisco.com
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol ils 389
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
names
access-list inside_outbound_nat0_acl permit ip 10.1.3.0 255.255.255.0 10.1.2.0
access-list inside_outbound_nat0_acl permit ip any any
access-list outside_cryptomap_20 permit ip 10.1.3.0 255.255.255.0 10.1.2.0 255.
access-list outside_cryptomap_40 permit ip any any
access-list 101 permit ip any any
pager lines 24
logging on
interface ethernet0 10baset
interface ethernet1 100full
interface ethernet2 100full shutdown

```

```
icmp permit any outside
icmp permit any inside
mtu outside 1500
mtu inside 1500
mtu intf2 1500
ip address outside 172.23.59.230 255.255.0.0 pppoe
ip address inside 10.1.3.1 255.255.255.0
ip address intf2 127.0.0.1 255.255.255.0
multicast interface inside
ip audit info action alarm
ip audit attack action alarm
no failover
failover timeout 0:00:00
failover poll 15
failover ip address outside 0.0.0.0
failover ip address inside 0.0.0.0
failover ip address intf2 0.0.0.0
pdm location 10.1.2.1 255.255.255.255 outside
pdm location 10.1.2.0 255.255.255.0 outside
pdm logging alerts 100
pdm history enable
arp timeout 14400
global (inside) 6 192.168.1.2-192.168.1.3
global (inside) 3 192.168.4.1
nat (inside) 0 access-list inside_outbound_nat0_acl
access-group 101 in interface outside
route outside 0.0.0.0 0.0.0.0 172.23.59.225 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h323 0:05:00 s0
timeout uauth 0:05:00 absolute
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
aaa-server LOCAL protocol local
http server enable
http 0.0.0.0 0.0.0.0 outside
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
floodguard enable
sysopt connection permit-ipsec
no sysopt route dnat
crypto ipsec transform-set ESP-DES-SHA esp-des esp-sha-hmac
crypto map outside_map 20 ipsec-isakmp
crypto map outside_map 20 match address outside_cryptomap_20
crypto map outside_map 20 set peer 172.23.59.231
crypto map outside_map 20 set transform-set ESP-DES-SHA
crypto map outside_map 40 ipsec-isakmp
crypto map outside_map 40 match address outside_cryptomap_40
crypto map outside_map 40 set peer 123.5.5.5
isakmp key ***** address 172.23.59.231 netmask 255.255.255.255 no-xauth no-c
isakmp peer fqdn no-xauth no-config-mode
isakmp policy 20 authentication pre-share
isakmp policy 20 encryption des
isakmp policy 20 hash sha
isakmp policy 20 group 2
isakmp policy 20 lifetime 86400
isakmp policy 40 authentication rsa-sig
isakmp policy 40 encryption 3des
isakmp policy 40 hash sha
isakmp policy 40 group 2
isakmp policy 40 lifetime 86400
telnet timeout 5
ssh timeout 5
```

```
terminal width 80
Cryptochecksum:4d600490f46b5d335c0fbf2eda0015a2
: end
```

## show startup-config

Display the PIX Firewall startup configuration. (Privileged mode.)

| Show command options       | Show command output  |
|----------------------------|--|
| <b>show startup-config</b> | Displays the configuration of the PIX Firewall at startup. |

### Syntax Description

**startup-config** The configuration present at startup on the PIX Firewall.

### Usage Guidelines

The **show startup-config** command displays the startup configuration of the PIX Firewall. The keyword **startup-config** is used to match Cisco IOS software command. The **show startup-config** command output is the same as the pre-existing PIX Firewall **show configure** command. The **show startup-config** command is not needed for PDM but is provided for compatibility with Cisco IOS software.

The **startup-config** keyword can be used only in the **show startup-config** command. It cannot be used with **no** or **clear**, or as a standalone command. If it is, the CLI treats it as a non-supported command. Also, for this reason, when **?**, **no ?**, or **clear ?** are entered, a **startup-config** option is not listed in the command list.

### Examples

The following is sample output from the **show startup-config** command:

```
pixfirewall# show startup-config
: Saved
: Written by enable_15 at 17:14:09.092 UTC Tue Apr 9 2002
PIX Version 6.2(0)227
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 intf2 security10
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname pixdoc515
domain-name cisco.com
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol ils 389
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
names
access-list inside_outbound_nat0_acl permit ip 10.1.3.0 255.255.255.0 10.1.2.0
access-list inside_outbound_nat0_acl permit ip any any
access-list outside_cryptomap_20 permit ip 10.1.3.0 255.255.255.0 10.1.2.0 255.
access-list outside_cryptomap_40 permit ip any any
```

```
access-list 101 permit ip any any
pager lines 24
logging on
interface ethernet0 10baset
interface ethernet1 100full
interface ethernet2 100full shutdown
icmp permit any outside
icmp permit any inside
mtu outside 1500
mtu inside 1500
mtu intf2 1500
ip address outside 172.23.59.230 255.255.0.0 pppoe
ip address inside 10.1.3.1 255.255.255.0
ip address intf2 127.0.0.1 255.255.255.0
multicast interface inside
ip audit info action alarm
ip audit attack action alarm
no failover
failover timeout 0:00:00
failover poll 15
failover ip address outside 0.0.0.0
failover ip address inside 0.0.0.0
failover ip address intf2 0.0.0.0
pdm location 10.1.2.1 255.255.255.255 outside
pdm location 10.1.2.0 255.255.255.0 outside
pdm logging alerts 100
pdm history enable
arp timeout 14400
global (inside) 6 192.168.1.2-192.168.1.3
global (inside) 3 192.168.4.1
nat (inside) 0 access-list inside_outbound_nat0_acl
access-group 101 in interface outside
route outside 0.0.0.0 0.0.0.0 172.23.59.225 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h323 0:05:00 s0
timeout uauth 0:05:00 absolute
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
aaa-server LOCAL protocol local
http server enable
http 0.0.0.0 0.0.0.0 outside
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
floodguard enable
sysopt connection permit-ipsec
no sysopt route dnat
crypto ipsec transform-set ESP-DES-SHA esp-des esp-sha-hmac
crypto map outside_map 20 ipsec-isakmp
crypto map outside_map 20 match address outside_cryptomap_20
crypto map outside_map 20 set peer 172.23.59.231
crypto map outside_map 20 set transform-set ESP-DES-SHA
crypto map outside_map 40 ipsec-isakmp
crypto map outside_map 40 match address outside_cryptomap_40
crypto map outside_map 40 set peer 123.5.5.5
isakmp key ***** address 172.23.59.231 netmask 255.255.255.255 no-xauth no-c
isakmp peer fqdn no-xauth no-config-mode
isakmp policy 20 authentication pre-share
isakmp policy 20 encryption des
isakmp policy 20 hash sha
isakmp policy 20 group 2
isakmp policy 20 lifetime 86400
isakmp policy 40 authentication rsa-sig
```

## show tech-support

```
isakmp policy 40 encryption 3des
isakmp policy 40 hash sha
isakmp policy 40 group 2
isakmp policy 40 lifetime 86400
telnet timeout 5
ssh timeout 5
```

# show tech-support

View information to help a support analyst. (Privileged mode.)

| Show command options     | Show command output   |
|--------------------------|---|
| <b>show tech-support</b> | Displays information that technical support analysts need to help diagnose PIX Firewall problems. |

### Syntax Description

**tech-support** The data used for diagnosis by technical support analysts.

### Usage Guidelines

The **show tech-support** command lists information that 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 from the **show tech-support** command:

```
show tech-support
PIX Version 6.0(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.)

| Configure with the command... | Remove with the command... |
|-------------------------------|----------------------------|
| N/A                           | <b>clear traffic</b>       |

| Show command options | Show command output   |
|----------------------|---|
| <b>show traffic</b>  | Displays the number of packets and bytes moving through each interface. |

### Syntax Description

**traffic** The packets and bytes moving through an interface.

**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 from 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/clear uauth

Delete all authorization caches for a user. (Privileged mode.)

| Display with the command... | Clear with the command...            |
|-----------------------------|--------------------------------------|
| <b>show uauth</b>           | <b>clear uauth</b> <i>[username]</i> |

| Show command options                | Show command output   |
|-------------------------------------|---|
| <b>show uauth</b> <i>[username]</i> | 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. |

**Syntax Description**

|                 |  |
|-----------------|--|
| <i>username</i> | Clear or view user authentication information by username. |
|-----------------|--|

**Usage Guidelines**

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 **clear uauth** command deletes one user's, or all users,' AAA authorization and authentication caches, which forces the user or users to reauthenticate the next time they create a connection. The **show uauth** command also lists CiscoSecure 2.1 and later idletime and timeout values, which can be set for different user groups.

This command is used in conjunction with the **timeout** command.

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 proxies the connection. This means that once you are authorized to access a website, 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. 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.

## Examples

The following is sample output from the **show uauth** command when no users are authenticated and one user authentication is in progress:

```
pixfirewall(config)# show uauth
                Current      Most Seen
Authenticated Users      0          0
Authen In Progress      0          1
```

The following is sample output from the **show uauth** command when three users are authenticated and authorized to use services through the PIX Firewall:

```
pixfirewall(config)# 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
```

## Related Commands

- [aaa authorization](#)
- [timeout](#)

# show version

View the PIX Firewall operating information. (Unprivileged mode.)

| Show command options | Show command output  |
|----------------------|--|
| <b>show version</b>  | Displays the PIX Firewall unit's software version, operating time since last reboot, processor type, Flash memory type, interface boards, serial number (BIOS ID), activation key value, and timestamp for when the configuration was last modified. |

|                           |                |  |
|---------------------------|----------------|--|
| <b>Syntax Description</b> | <b>version</b> | The PIX Firewall software version, hardware configuration, license key, and related uptime data. |
|---------------------------|----------------|--|

**Usage Guidelines**

Use the **show version** command to display 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.

The serial number listed with the **show version** command in PIX Firewall software version 5.3 and higher 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.

Throughput Limited indicates that the speed of the PIX Firewall interface is limited due to platform or version restrictions. ISAKMP peers Limited indicates that the number of IPsec peers is limited due to platform restrictions.

**Note**

The uptime value in the output from the **show version** command indicates how long a failover set has been running. If one unit stops running, the uptime value will continue to increase as long as the other unit continues to operate.

For PIX Firewall software version 6.2 and higher, the **show version** command output appears as follows:

```
Running Activation Key: activation-key-four-tuple
```

to indicate the activation key that is currently running PIX Firewall image.

In the following examples, the amount of Flash memory (2 MB or 16 MB) is identified as follows:

```
i28F020          512 kB
AT29C040A       2 MB
atmel           2 MB
i28F640J5       8 MB - PIX 506
                16 MB - all other PIXes
strata          16 MB
E28F128J3      16 MB
```

**Examples**

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

```
pixfirewall(config)# show version

Cisco PIX Firewall Version 6.2(1)
Cisco PIX Device Manager Version 2.0(1)

Compiled on Wed 17-Apr-02 21:18 by morlee

pixdoc515 up 9 days 3 hours

Hardware:   PIX-515, 64 MB RAM, CPU Pentium 200 MHz
Flash i28F640J5 @ 0x300, 16MB
BIOS Flash AT29C257 @ 0xffffd8000, 32KB

0: ethernet0: address is 0050.54ff.3772, irq 10
1: ethernet1: address is 0050.54ff.3773, irq 7
2: ethernet2: address is 00d0.b792.409d, irq 11
Licensed Features:
Failover:           Enabled
```

## show xlate/clear xlate

```

VPN-DES:          Enabled
VPN-3DES:         Enabled
Maximum Interfaces: 6
Cut-through Proxy: Enabled
Guards:           Enabled
URL-filtering:    Enabled
Inside Hosts:     Unlimited
Throughput:       Unlimited
IKE peers:        Unlimited

```

```

Serial Number: 480221353 (0x1c9f98a9)
Running Activation Key: 0x36df4255 0x246dc5fc 0x39d2ec4d 0x09f6288f
Configuration last modified by enable_15 at 12:15:28.311 UTC Wed May 1 2002
pixfirewall(config)#

```

## show xlate/clear xlate

View or clear translation slot information. (Privileged mode.)

| Display with the command... | Clear with the command...  |
|-----------------------------|--|
| <code>show xlate</code>     | <code>clear xlate [global   local ip1 [-ip2] [netmask mask]] lport   gport port [-port]] [interface if1 [,if2] [,ifn]] [state static [,dump] [,portmap] [,norandomseq] [,identity]]</code> |

| Show command options   | Show command output                                  |
|--|--|
| <code>show xlate [detail] [global   local ip1 [-ip2] [netmask mask]] lport   gport port [-port]] [interface if1 [,if2] [,ifn]] [state static [,dump] [,portmap] [,norandomseq] [,identity]] [debug] [count]</code> | Displays the contents of only the translation slots. |

### Syntax Description

|   |  |
|---|--|
| <b>detail</b>                                     | If specified, displays translation type and interface information.   |
| <b>[global   local ip1 [-ip2] [netmask mask]]</b> | Display active translations by global IP address or local IP address using the network mask to qualify the IP addresses.   |
| <b>interface if1 [,if2] [,ifn]</b>                | Display active translations by interface.  |
| <b>lport   gport port [-port]</b>                 | Display active translations by local and global port specifications. See “Ports” in Chapter 2, “Using PIX Firewall Commands” for a list of valid port literal names.   |
| <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> ). |
| <b>debug</b>                                      | Display translation type and interface information.  |
| <b>count</b>                                      | Display the number of active translations.   |

**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.

The **show xlate detail** command displays the following information:

**{ICMP|TCP|UDP} PAT from *interface:real-address/real-port* to *interface:mapped-address/mapped-port* flags *translation-flags***

**NAT from *interface:real-address/real-port* to *interface:mapped-address/mapped-port* flags *translation-flags***

The translation flags are defined in [Table 8-3](#).

**Table 8-3 Translation Flags**

| Flag | Description                                    |
|------|--|
| s    | static translation slot                        |
| d    | dump translation slot on next cleaning cycle   |
| r    | portmap translation (Port Address Translation) |
| n    | no randomization of TCP sequence number        |
| o    | outside address translation                    |
| i    | inside address translation                     |
| D    | DNS A RR rewrite                               |
| I    | identity translation from <b>nat 0</b>         |

**Examples**

The following is sample output from the **show xlate** command with three active Port Address Translations (PATs):

```
pixfirewall(config)# show xlate
3 in use, 3 most used
PAT Global 192.150.49.1(0) Local 10.1.1.15 ICMP id 340
PAT Global 192.150.49.1(1024) Local 10.1.1.15(1028)
PAT Global 192.150.49.1(1024) Local 10.1.1.15(516)
```

The following is sample output from the **show xlate detail** command with three active Port Address Translations (PATs):

```
pixfirewall(config)# show xlate detail
3 in use, 3 most used
Flags: D - DNS, d - dump, I - identity, i - inside, n - no random,
       o - outside, r - portmap, s - static
TCP PAT from inside:10.1.1.15/1026 to outside:192.150.49.1/1024 flags ri
UDP PAT from inside:10.1.1.15/1028 to outside:192.150.49.1/1024 flags ri
ICMP PAT from inside:10.1.1.15/21505 to outside:192.150.49.1/0 flags ri
```

The first entry is a TCP Port Address Translation for host-port (10.1.1.15, 1025) on the inside network to host-port (192.150.49.1, 1024) on the outside network. The flag "r" denotes the translation is a Port Address Translation. The "i" flag denotes that the translation applies to the inside address-port.

The second entry is a UDP Port Address Translation for host-port (10.1.1.15, 1028) on the inside network to host-port (192.150.49.1, 1024) on the outside network. The flag "r" denotes the translation is a Port Address Translation. The "i" flags denotes that the translation applies to the inside address-port.

The third entry is an ICMP Port Address Translation for host-ICMP-id (10.1.1.15, 21505) on the inside network to host-ICMP-id (192.150.49.1, 0) on the outside network. The flag "r" denotes the translation is a Port Address Translation. The "i" flags denotes that the translation applies to the inside address-ICMP-id.

The inside address fields appear as source addresses on packets traversing from the more secure interface to the less secure interface. Conversely, they appear as destination addresses on packets traversing from the less secure interface to the more secure interface.

The following is sample output from 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
```

### Related Commands

- [show conn](#)
- [timeout](#)
- [show uauth/clear uauth](#)

## shun

The **shun** command enables a dynamic response to an attacking host by preventing new connections and disallowing packets from any existing connection. (Configuration Mode.)

| Configure with the command...  | Remove with the command...   |
|--|--|
| <b>shun</b> <i>src_ip</i> [ <i>dst_ip sport dport</i> [ <i>protocol</i> ]] | <b>no shun</b> <i>src_ip</i> [ <i>dst_ip sport dport</i> [ <i>protocol</i> ]]<br><br><b>clear shun</b> [ <i>statistics</i> ] |

| Show command options                                   | Show command output   |
|--|---|
| <b>show shun</b> [ <i>src_ip</i>   <i>statistics</i> ] | Displays all shuns currently enabled in the exact format specified. |

### Syntax Description

|                 |  |
|-----------------|--|
| <b>clear</b>    | Disable all shuns currently enabled and clears shun statistics. Specifying statistics only clears the counters for that interface. |
| <i>dport</i>    | The destination port of the connection causing the shun.   |
| <i>dst_ip</i>   | The address of the of the target host.   |
| <b>no</b>       | Disable a shun based on <i>src_ip</i> , the actual address used by the PIX Firewall for shun lookups.                              |
| <i>protocol</i> | The optional IP protocol, such as UDP or TCP.  |
| <b>shun</b>     | Enable a blocking function (shun) based on <i>src_ip</i> .   |

|                   |   |
|-------------------|---|
| <i>sport</i>      | The source port of the connection causing the shun. |
| <i>src_ip</i>     | The address of the attacking host.                  |
| <i>statistics</i> | Clear only interface counters.                      |

### Usage Guidelines

The **shun** command applies a blocking function to the interface receiving the attack. Packets containing the IP source address of the attacking host will be dropped and logged until the blocking function is removed manually or by the Cisco IDS master unit. No traffic from the IP source address will be allowed to traverse the PIX Firewall unit and any remaining connections will time out as part of the normal architecture. The blocking function of the **shun** command is applied whether or not a connection with the specified host address is currently active.

If the **shun** command is used only with the source IP address of the host, then the other defaults will be 0. No further traffic from the offending host will be allowed.

Because the **shun** command is used to block attacks dynamically, it is not displayed in your PIX Firewall configuration.

### Examples

In the following example, the offending host (10.1.1.27) makes a connection with the victim (10.2.2.89) with TCP. The connection in the PIX Firewall connection table reads:

```
10.1.1.27, 555-> 10.2.2.89, 666 PROT TCP
```

If the **shun** command is applied in the following way:

```
shun 10.1.1.27 10.2.2.89 555 666 tcp
```

The preceding command would delete the connection from the PIX Firewall connection table, and it would also prevent packets from 10.1.1.27 from going through the PIX Firewall. The offending host can be inside or outside of the PIX Firewall.

The following is sample output of the **show shun** command with the **shun** command applied to the outside interface:

```
outside=ON, cnt=4, time=(0:04:13)
```

The first value indicates if the **shun** command is applied to the interface, the second value (**cnt**) indicates the number of packets that have been dropped since the **shun** command was applied. The third value (**time**) indicates the elapsed time since the **shun** command was applied to the interface.

## snmp-server

Provide PIX Firewall event information through SNMP. (Configuration mode.)

| Configure with the command...   | Disable with the command...                                |
|---|--|
| <b>snmp-server community</b> <i>key</i>   | <b>no snmp-server community</b> <i>key</i>                 |
| <b>snmp-server</b> { <b>contact</b>   <b>location</b> } <i>text</i>                     | <b>no snmp-server</b> { <b>contact</b>   <b>location</b> } |
| <b>snmp-server host</b> [ <i>if_name</i> ] <i>ip_addr</i> [ <b>trap</b>   <b>poll</b> ] | <b>no snmp-server</b> [ <i>if_name</i> ] <i>ip_addr</i>    |
| <b>snmp-server enable traps</b>   | <b>no snmp-server enable traps</b>                         |
| N/A   | <b>clear snmp-server</b>                                   |

| Show command options          | Show command output              |
|-------------------------------|----------------------------------|
| <code>show snmp-server</code> | Displays the SNMP configuration. |

**Syntax Description**

|                             |   |
|-----------------------------|---|
| <b>community</b> <i>key</i> | 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.<br><br>The <i>key</i> is a case-sensitive value up to 32 characters in length. Spaces are not permitted. The default is <b>public</b> if <i>key</i> is not set. Consequently, it is important to specify a (new) value for <i>key</i> for security reasons. |
| <b>contact</b> <i>text</i>  | 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.  |
| <b>enable traps</b>         | Enable or disable sending log messages as SNMP trap notifications.  |
| <b>host</b>                 | 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 32 SNMP management stations.   |
| <i>if_name</i>              | The interface name where the SNMP management station resides.   |
| <i>ip_addr</i>              | The IP address of a host to which SNMP traps should be sent and/or from which the SNMP requests come.   |
| <b>location</b> <i>text</i> | 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.   |
| <b>trap   poll</b>          | Specify whether traps, polls, or both are acted upon. Use with these parameters: <ul style="list-style-type: none"> <li>• <b>trap</b>—Only traps will be sent. This host will not be allowed to poll.</li> <li>• <b>poll</b>—Traps will not be sent. This host will be allowed to poll.</li> </ul> The default allows both traps and polls to be acted upon.  |

**Usage Guidelines**

Use the **snmp-server** command to identify site, management station, community string, and user information.

**Note**

In the **snmp-server community** *key* command, the default value for *key* is **public**. Consequently, it is important to specify a (new) value for *key* for security reasons.

The **clear snmp-server** and **no snmp-server** commands disable the SNMP commands in the configuration as follows:

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

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 displays in SNMP event traps sent from the PIX Firewall. The OIDs for the PIX Firewall platforms are listed in [Table 8-4](#).

**Table 8-4 System OID in PIX Firewall Platforms**

| PIX Firewall Platform | System OID           |
|-----------------------|----------------------|
| PIX 506               | .1.3.6.1.4.1.9.1.389 |
| PIX 506E              | .1.3.6.1.4.1.9.1.450 |
| PIX 515               | .1.3.6.1.4.1.9.1.390 |
| PIX 515E              | .1.3.6.1.4.1.9.1.451 |
| PIX 520               | .1.3.6.1.4.1.9.1.391 |
| PIX 525               | .1.3.6.1.4.1.9.1.392 |
| PIX 535               | .1.3.6.1.4.1.9.1.393 |
| Others                | .1.3.6.1.4.1.9.1.227 |

Use the **trap** and **poll** command options to configure hosts to participate only in specific SNMP activities. Poll responses and traps are sent only to the configured entities. Hosts configured with the **trap** command option will have traps sent to them, but will not be allowed to poll. Hosts configured with the **poll** command option will be allowed to poll, but will not have traps sent to them. Refer to the *Cisco PIX Firewall and VPN Configuration Guide* for more information on how to access and monitor the PIX Firewall using SNMP traps.

Accessibility to PIX Firewall Management Information Bases (MIBs) is based on configuration, MIB support, and authentication based on the community string. Unsuccessful polling attempts, except for failed community string authentication, are not logged or otherwise indicated. Community authentication failures result in a trap where applicable.

### 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, Cisco Memory Pool MIB, Cisco Process MIB provide the following PIX Firewall information through SNMP:

- Buffer usage from the **show block** command
- Connection count from the **show conn** command
- CPU usage through the **show cpu usage** command
- Failover status
- Memory usage from the **show memory** command

### Receiving SNMP Requests from an SNMP Management Station

To receive SNMP requests from a management station, perform the following steps:

- 
- 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.
- 

#### Defaults

If you do not specify an option, the **snmp-server host** command behaves as in previous versions. The polling is permitted from all configured hosts on the affected interface. Traps are sent to all configured hosts on the affected interface.

#### 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 from 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 through Secure Shell (SSH). (Configuration mode.)

| Configure with the command...   | Remove with the command...  |
|---|---|
| <b>ssh</b> <i>ip_address</i> [ <i>netmask</i> ] [ <i>interface_name</i> ] | <b>no ssh</b> <i>ip_address</i> [ <i>netmask</i> ]<br>[ <i>interface_name</i> ] |
| <b>ssh timeout</b> <i>mm</i>  | N/A   |
| N/A   | <b>ssh disconnect</b> <i>session_id</i>   |
| N/A   | <b>clear ssh</b>  |

| Show command options                                     | Show command output  |
|--|--|
| <b>show ssh</b> [ <b>sessions</b> [ <i>ip_address</i> ]] | Displays active (all or host-specific) SSH sessions on the PIX Firewall. |
| <b>show ssh timeout</b>                                  | Displays SSH timeout information.  |

**Syntax Description**

|                       |   |
|-----------------------|---|
| <i>interface_name</i> | PIX Firewall interface name on which the host or network initiating the SSH connection resides.   |
| <i>ip_address</i>     | IP address of the host or network authorized to initiate an SSH connection to the PIX Firewall.   |
| <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>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>session_id</i>     | SSH session ID number, viewable with 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. After generating the RSA key-pair, save the key-pair using the **ca save all** command. 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 the 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 protocol version number that the SSH client supports. 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 login username that has been authenticated for the session. The "pix" username appears when non-AAA authentication is used.

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, 315005, and 315011 were added for SSH. Refer to *Cisco PIX Firewall System Log Messages* 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 website:

<http://hp.vector.co.jp/authors/VA002416/teraterm.html>

The TTSSH security enhancement for Tera Term Pro is available at the following website:

<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\Ttempo folder.

- Linux, Solaris, OpenBSD, AIX, IRIX, HP/UX, FreeBSD, and NetBSD—download the SSH v1.x client from the following website:

<http://www.openssh.com>

- Macintosh (international users only)—download the Nifty Telnet 1.1 SSH client from the following website:

<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 the **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.

### 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 host name 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 perform 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
```

### Related Commands

- [aaa accounting](#)
- [ca](#)
- [domain-name](#)

- [hostname](#)
- [passwd](#)

## static

Configure a persistent one-to-one address translation rule by mapping a local IP address to a global IP address. This is also known as Static Port Address Translation (Static PAT). (Configuration mode.)

| Configure with the command...  | Remove with the command...  |
|--|---|
| <b>static</b> [( <i>prenat_interface</i> , <i>postnat_interface</i> )]<br>{ <b>mapped_address</b>   <b>interface</b> }<br><i>real_address</i> [ <b>dns</b> ] [ <b>netmask</b> <i>mask</i> ]<br>[ <b>norandomseq</b> ] [ <i>connection_limit</i> ]<br>[ <i>em_limit</i> ]]                                      | <b>no static</b> [( <i>prenat_interface</i> ,<br><i>postnat_interface</i> )] { <b>mapped_address</b>  <br><b>interface</b> } <i>real_address</i> [ <b>dns</b> ] [ <b>netmask</b><br><i>mask</i> ] [ <b>norandomseq</b> ] [ <i>max_conns</i><br>[ <i>em_limit</i> ]]   |
| <b>static</b> [( <i>internal_if_name</i> ,<br><i>external_if_name</i> )] { <b>tcp</b>  <br><b>udp</b> } { <i>global_ip</i>   <b>interface</b> } <i>global_port</i><br><i>local_ip</i> <i>local_port</i> [ <b>netmask</b><br><i>mask</i> ] [ <i>max_conns</i> ] [ <i>emb_limit</i> ]<br>[ <b>norandomseq</b> ]] | <b>no static</b> [( <i>internal_if_name</i> ,<br><i>external_if_name</i> )] { <b>tcp</b>  <br><b>udp</b> } { <i>global_ip</i>   <b>interface</b> } <i>global_port</i><br><i>local_ip</i> <i>local_port</i> [ <b>netmask</b><br><i>mask</i> ] [ <i>max_conns</i> ] [ <i>emb_limit</i><br>[ <b>norandomseq</b> ]] |

| Show command options | Show command output                            |
|----------------------|--|
| <b>show static</b>   | Displays static commands in the configuration. |

### Syntax Description

|                                       |   |
|---------------------------------------|---|
| <b>dns</b>                            | Specifies that DNS replies that match the xlate are translated.   |
| <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.   |
| <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 Port Address Translation (PAT) IP address. The IP address on the lower security level interface you are accessing.  |
| <b>interface</b>                      | Specifies to overload the global address from interface.  |
| <i>internal_if_name</i>               | The internal network interface name. The higher 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.  |
| <i>mapped_address</i>                 | The address <i>real_address</i> is translated into.   |
| <i>mapped_port</i>                    | The port <i>real_port</i> is translated into.   |
| <i>mask</i> or<br><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 each translation at the same time.   |
| <b>netmask</b>           | Reserve word required before specifying the network mask.  |
| <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.                      |
| <i>postnat_interface</i> | The outside interface when <i>prenat_interface</i> is the inside interface. However, if the outside interface is used for <i>prenat_interface</i> , then the translation is applied to the outside address and the <i>postnat_interface</i> is the inside interface. |
| <i>prenat_interface</i>  | Usually the inside interface, in which case the translation is applied to the inside address.  |
| <i>real_address</i>      | The address to be mapped.  |
| <i>real_port</i>         | The port to be mapped.   |

### Usage Guidelines

The **static** command creates a persistent, one-to-one address translation rule (called a static translation slot or "xlate"). This translation can be between a local IP address and a global IP address (static NAT) or between ports (static PAT). Additionally, the PIX Firewall dynamically creates a secondary xlate using the global address in the static command.



#### Note

When changing static assignments, you may need to use the clear xlate command to clear the old translation and to enable the new translation.

The following example redirects the FTP service from address 198.168.1.1 to inside host 10.1.1.1, where the address translation slots (xlates) necessary for FTP data transfer are automatically created from the global address 192.168.1.1 by the **fixup** application inspection:

```
static (inside, outside) tcp 192.168.1.1 ftp 10.1.1.1 ftp
fixup protocol ftp 21
```

For an external host to initiate traffic to an inside host, a static translation rule needs to exist for the inside host; this can also be done using a **nat 0 access-list** address translation rule. Without the persistent translation rule, the translation cannot occur.

You can use the **static** and **access-list** commands when you are accessing the 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.

### Static Port Address Translation (Static PAT)

Static PAT is a many-to-one port mapping that is constant over time. For example, static PAT lets you redirect inbound TCP and UDP services. Using the **static** command **interface** option, you can use Static PAT to permit external hosts access TCP or UDP services residing on an internal host. (As always, though, an access list should also be in place to control access to the internal host.)

Static PAT supports all applications that are supported by (regular) PAT, including the same application constraints.



#### Note

PIX Firewall Version 6.2 introduces support for PAT and Static PAT of H.323 application traffic; PAT is not supported for H.323 in earlier versions.

The Telnet port 23 and PFM port 1467 of the PIX Firewall interface cannot be used for Static PAT because the PIX Firewall requires traffic to these ports be protected by IPSec.

The following examples enable static port address translations (Static PATs) for the following services, interfaces, and hosts:

- Telnet to the PIX Firewall outside interface to be redirected inside host 10.1.1.15:

```
static (inside, outside) tcp interface telnet 10.1.1.15 telnet
```

- FTP to the PIX Firewall outside interface to be redirected inside host 10.1.1.30:

```
static (inside, outside) tcp interface ftp 10.1.1.30 ftp
```

- DNS to the PIX Firewall outside interface to be redirected inside host 10.1.1.30:

```
static (inside, outside) udp interface domain 10.1.1.30 domain
```

### TCP Intercept Feature

Prior to version 5.3, 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 rules for which command to use with an interface is summarized in Table 8-5. Table 8-5 assumes that the security levels are 40 for dmz1 and 60 for dmz2.

**Table 8-5 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>       |
| dmz2                | inside            | <b>static</b>    |
| outside             | dmz1              | <b>static</b>    |
| outside             | dmz2              | <b>static</b>    |
| outside             | inside            | <b>static</b>    |

### Using Statics

For the interface names in the **static** command, always specify the highest security level interface name first, and then the lower security level interface name. However, the IP addresses are specified in the opposite order because the first IP address you specify is for the lower security level interface, and the second IP address is for the higher security level interface. The way to remember this is as follows:

```
static (if_name_high, if_name_low) ip_address_low ip_address_high
```

where the highest security level interface is an inside interface, and the lowest security level interface is an outside interface.

If you do not want an address translation, the format of the **static** command is as follows:

```
static (if_name_high, if_name_low) ip_address ip_address
```

where the interface IP addresses are the same.

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, use the **static** command as follows:

```
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, use the **static** command as follows:

```
static (inside,dmz1) dmz1_ip_address inside_ip_address netmask mask
```

To access the inside from the dmz2 interface, use the **static** command as follows:

```
static (inside,dmz2) dmz2_ip_address inside_ip_address netmask mask
```

To access the dmz2 interface from the dmz1 interface, use the **static** command as follows:

```
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 the following 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** command 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 NAT disabled, the **static** command has a different sense of logic. With NAT disabled, addresses on both sides of the PIX Firewall are registered addresses. Between interfaces, addresses must be on different subnets that you control with subnetting. See the *Cisco PIX Firewall and VPN Configuration Guide* for more information about subnetting.

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.

**Examples**

The example that follows creates a **static** command and then permits users to call in through H.323 using Intel Internet Phone, 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 255.255.255.224
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.

**Related Commands**

- [access-list](#)

## syslog

Enable syslog message facility. Obsolete command replaced by the [logging](#) command. (Privileged mode.)

See the [logging](#) command for more information. The **syslog** command is available for backward compatibility.

## sysopt

Change PIX Firewall system options. (Configuration mode.)

| Configure with the command...                                     | Remove with the command...   |
|---|--|
| <b>sysopt connection permit-pptp   permit-l2tp   permit-ipsec</b> | <b>no sysopt connection permit-pptp   permit-l2tp   permit-ipsec</b> |
| <b>sysopt connection tcpmss <i>bytes</i></b>                      | <b>no sysopt connection tcpmss <i>bytes</i></b>                      |
| <b>sysopt connection timewait</b>                                 | <b>no sysopt connection timewait</b>                                 |
| <b>sysopt ipsec pl-compatible</b>                                 | <b>no sysopt ipsec pl-compatible</b>                                 |
| <b>sysopt nodnsalias inbound   outbound</b>                       | <b>no sysopt nodnsalias inbound   outbound</b>                       |
| <b>sysopt noproxyarp <i>if_name</i></b>                           | <b>no sysopt noproxyarp <i>if_name</i></b>                           |
| <b>sysopt radius ignore-secret</b>                                | <b>no sysopt radius ignore-secret</b>                                |
| <b>sysopt route dnat</b>  | <b>no sysopt route dnat</b>  |
| <b>sysopt security fragguard</b>                                  | <b>no sysopt security fragguard</b>                                  |

| Configure with the command...        | Remove with the command...              |
|--------------------------------------|---|
| <b>sysopt uauth allow-http-cache</b> | <b>no sysopt uauth allow-http-cache</b> |
| N/A                                  | <b>clear sysopt</b>                     |

| Show command options | Show command output                                       |
|----------------------|---|
| <b>show sysopt</b>   | Displays the <b>sysopt</b> commands in the configuration. |

### Syntax Description

|                                       |  |
|---------------------------------------|--|
| <b>connection permit-ipsec</b>        | Implicitly permit any packet that came from an IPsec tunnel and bypass the checking of an associated <b>access-list</b> , <b>conduit</b> , or <b>access-group</b> command statement for IPsec connections.           |
| <b>connection permit-l2tp</b>         | Implicitly permit any packet that came from an L2TP/IPsec tunnel and bypass the checking of an associated <b>access-list</b> , <b>conduit</b> , or <b>access-group</b> command statement for L2TP/IPsec connections. |
| <b>connection permit-pptp</b>         | Allow PPTP traffic to bypass <b>conduit</b> or <b>access-list</b> command statement checking.  |
| <b>connection tcpmss</b> <i>bytes</i> | 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>ipsec pl-compatible</b>            | Enable IPsec packets to bypass the PIX Firewall unit's NAT and ASA features and allows incoming IPsec packets to terminate on the inside interface.  |
| <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>radius ignore-secret</b>           | Ignore authenticator key to avoid retransmit caveat.   |
| <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>uauth allow-http-cache</b>         | Allows the web browser to supply a username and password from its cache for AAA authentication.  |

### 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.

There is no need to enter the **sysopt connection permit-l2tp** command if the **sysopt connection permit-ipsec** command is present.

#### **sysopt connection permit-ipsec**

Use the **sysopt connection permit-ipsec** command in IPsec configurations to permit IPsec traffic to pass through the PIX Firewall without a check of **conduit** or **access-list** command statements.

An **access-list** or **conduit** command statement must be available for inbound sessions.

By default, any inbound session must be explicitly permitted by a **conduit** or **access-list** command statement. With IPsec protected traffic, the secondary access list check could be redundant. To enable IPsec authenticated/cipher inbound sessions to always be permitted, use the **sysopt connection permit-ipsec** command.

If both the **sysopt ipsec pl-compatible** command and the **sysopt connection permit-ipsec** command are used within your configuration, the **sysopt ipsec pl-compatible** command will take precedence.

If the **sysopt connection permit-ipsec** command is not configured, you must explicitly configure an **access-list** command statement to permit IPsec traffic to traverse the PIX Firewall.

The **no sysopt connection permit-ipsec** command disables the option.

#### **sysopt connection permit-pptp**

Let PPTP traffic bypass **conduit** and **access-list** command statement checking. Use the **vpdn** command to implement PPTP.

#### **sysopt connection permit-l2tp**

This command allows L2TP traffic to bypass conduit/access-list checking. Because L2TP traffic can only come from IPsec, the **sysopt connection permit-ipsec** command will allow L2TP traffic to pass as well.

#### **sysopt ipsec pl-compatible**



#### Note

The **sysopt ipsec pl-compatible** command provides a migration path for Private Link users from Private Link tunnels to IPsec tunnels.

The **sysopt ipsec pl-compatible** command enables the IPsec feature to simulate the Private Link feature supported in PIX Firewall version 4. The Private Link feature provides encrypted tunnels to be established across an unsecured network between Private-Link equipped PIX Firewall units. The **sysopt ipsec pl-compatible** command allows IPsec packets to bypass the NAT and ASA features and enables incoming IPsec packets to terminate on the sending interface.

The **sysopt ipsec pl-compatible** command is not available on a PIX 501.

The **no sysopt ipsec pl-compatible** command disables the option, which is off by default.



#### Note

When using the **sysopt ipsec pl-compatible** command, all PIX Firewall features, such as access list control, stateful inspection, and user authentication, are bypassed for IPsec packets only.

If both the **sysopt ipsec pl-compatible** command and the **sysopt connection permit-ipsec** command are used within your configuration, the **sysopt ipsec pl-compatible** command will take precedence.

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.

#### **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 data + 20 TCP + 20 IP + 24 AH + 24 ESP_CIPHER + 12 ESP_AUTH + 20 IP = 1500 bytes
```

1500 bytes is the MTU for Ethernet connections. We recommend that the default value of 1380 bytes be used for Ethernet. 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.

**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**

By default the PIX Firewall does not use the **timewait** option.

Use the **sysopt connection timewait** command to enable the **timewait** option when you have an end host application whose default TCP terminating sequence is a simultaneous close.

This is recommended because the default behavior of the PIX Firewall is to track the 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, based on the most common closing sequence, known as the normal close sequence. However, in a simultaneous close, both ends of the transaction initiate the closing sequence, as opposed to the normal close sequence where one end closes and the other end acknowledges prior to initiating its own closing sequence (see RFC 793). Thus, in a simultaneous close, the quick release forces one side of the connection to linger in the CLOSING state. Having 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. Using the **sysopt connection timewait** command creates a window for the simultaneous close down sequence to complete.

The **no sysopt connection timewait** command removes the **sysopt connection timewait** command from your configuration. In other words, if you enable the **timewait** option with the **sysopt connection timewait** command, you can disable it using the **no sysopt connection timewait** command.

**Note**

The **sysopt connection timewait** command requires more system resources than default processing and, when in use, may impact PIX Firewall performance. Noticeable performance impact is most likely when there is limited memory available, and when there is highly dynamic traffic 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**

ARP (Address Resolution Protocol) is a layer two protocol that resolves an IP address to a physical address, also called a Media Access Controller (MAC) address. A host sends an ARP request asking “Who is this IP?” The device owning the IP should reply with “Hey, I am the one, here's my MAC address.”

Proxy ARP refers to a gateway device, in this case, the firewall, “impersonating” an IP address and returning its own MAC address to answer an ARP request for another device.

The firewall builds a table from responses to ARP requests to map physical addresses to IP addresses. A periodic ARP function is enabled in the default configuration. The presence of entries in the ARP cache indicates that the firewall has network connectivity. The show arp command lists the entries in the ARP table. Usually, administrators do not need to manually manipulate ARP entries on the firewall. This is done only when troubleshooting or solving network connectivity problems.

The arp command is used to add a permanent entry for host on a network. If one host is exchanged for another host with the same IP address then the “clear arp” command can be used to clear the ARP cache on the PIX. 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 sysopt noproxyarp command is used to disable Proxy ARPs on an interface from the command-line interface. By default, the PIX Firewall responds to ARP requests directed at the PIX Firewall's interface IP addresses as well as to ARP requests for any static or global address defined on the PIX Firewall interface (which are proxy ARP requests).

The **sysopt noproxyarp** *if\_name* command lets you disable proxy ARP request responses on a PIX Firewall interface. However, this command does not disable (non-proxy) ARP requests on the PIX Firewall interface itself. Consequently, if you use the **sysopt noproxyarp** *if\_name* command, the PIX Firewall no longer responds to ARP requests for the addresses in the **static**, **global**, and **nat 0** commands for that interface but does respond to ARP requests for its interface IP addresses.

To disable Proxy ARPs on the inside interface:

```
sysopt noproxyarp inside
```

To enable Proxy ARPs on the inside interface:

```
no sysopt noproxyarp inside
```

#### **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 an **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. We recommend that packet fragmentation not be permitted on the network if at all possible.

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
```

In the following example, a PPTP client authenticates using MS-CHAP, 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 permit-ipsec**

The following is a minimal IPSec configuration to enable a session to be connected from host 172.21.100.123 to host 172.21.200.67 across an IPSec tunnel that terminates from peer 209.165.201.1 to peer 201.165.200.225.

With **sysopt connection permit-ipsec** and **access-list** command statements:

On peer 209.165.201.1:

```
static 172.21.100.123 172.21.100.123
access-list 10 permit ip host 172.21.200.67 host 172.21.100.123
crypto ipsec transform-set t1 esp-des esp-md5-hmac
crypto map mymap 10 ipsec-isakmp
crypto map mymap 10 match address 10
crypto map mymap 10 set transform-set t1
crypto map mymap 10 set peer 172.21.200.1
crypto map mymap interface outside
```

On peer 201.165.200.225:

```
static 172.21.200.67 172.21.200.67
access-list 10 permit ip host 172.21.100.123 host 172.21.200.67
crypto ipsec transform-set t1 esp-des esp-md5-hmac
crypto map mymap 10 ipsec-isakmp
crypto map mymap 10 match address 10
crypto map mymap 10 set transform-set t1
crypto map mymap 10 set peer 172.21.100.1
crypto map mymap interface outside
```

With **sysopt connection permit-ipsec** and without **conduit** command statements:

On peer 209.165.201.1:

```
static 172.21.100.123 172.21.100.123
access-list 10 permit ip host 172.21.200.67 host 172.21.100.123
crypto ipsec transform-set t1 esp-des esp-md5-hmac
crypto map mymap 10 ipsec-isakmp
crypto map mymap 10 match address 10
crypto map mymap 10 set transform-set t1
crypto map mymap 10 set peer 172.21.200.1
crypto map mymap interface outside
sysopt connection permit-ipsec
```

On peer 201.165.200.225:

```
static 172.21.200.67 172.21.200.67
access-list 10 permit ip host 172.21.100.123 host 172.21.200.67
crypto ipsec transform-set t1 esp-des esp-md5-hmac
crypto map mymap 10 ipsec-isakmp
crypto map mymap 10 match address 10
crypto map mymap 10 set transform-set t1
crypto map mymap 10 set peer 172.21.100.1
crypto map mymap interface outside
sysopt connection permit-ipsec
```

