



## S Commands

---

### service

Reset inbound connections. (Configuration mode.)

**service resetinbound**

**service resetoutside**

**show service**

**clear service**

---

#### Syntax Description

---

**resetinbound**    Reset inbound connections.

---

---

#### Usage Guidelines

The **service** command works with all inbound TCP connections to statics whose access lists or uauth (user authorization) do not allow inbound. One use is for resetting IDENT connections. If an inbound TCP connection is attempted and denied, you can use the **service resetinbound** command to return an RST (reset flag in the TCP header) to the source. Without the option, the PIX Firewall drops the packet without returning an RST.

For use with IDENT, the PIX Firewall sends a TCP RST to the host connecting inbound and stops the incoming IDENT process so that email outbound can be transmitted without having to wait for IDENT to time out. In this case, the PIX Firewall sends a syslog message stating that the incoming connection was a denied connection. Without **service resetinbound**, the PIX Firewall drops packets that are denied and generates a syslog message stating that the SYN was a denied connection. However, outside hosts keep retransmitting the SYN until the IDENT times out.

When an IDENT connection is timing out, you will notice that connections slow down. Perform a trace to determine that IDENT is causing the delay and then invoke the **service** command.

The **service resetinbound** command provides a safer way to handle an IDENT connection through the PIX Firewall. Ranked in order of security from most secure to less secure are these methods for handling IDENT connections:

1. Use the **service resetinbound** command.
2. Use the **established** command with the **permitto tcp 113** options.
3. Enter **static** and **access-list** command statements to open TCP port 113.

When using the **aaa** command, if the first attempt at authorization fails and a second attempt causes a timeout, use the **service resetinbound** command to reset the client that failed the authorization so that it will not retransmit any connections. An example authorization timeout message in Telnet follows:

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

---

## Examples

The following example shows use of the **service resetinbound** command:

```
service resetinbound
show service
service resetinbound
```

If you use the **resetoutside** command, the PIX Firewall actively resets denied TCP packets that terminate at the PIX Firewall 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 allows you to provide pre-configuration information to a new PIX Firewall, so you can then configure and monitor your PIX Firewall graphically using PDM. (Configuration Mode.)

## setup

```
Pre-configure PIX Firewall now through interactive prompts [yes]?
Enable Password [<use current password>]:
Clock (UTC)
  Year [system year]:
  Month [system month]:
  Day [system day]:
  Time [system time]:
Inside IP address:
Inside network mask:
Host name:
Domain name:
IP address of host running PIX Device Manager:
```

### Syntax Description

<b>setup</b>	Prompts for the basic operational information for the PIX Firewall if no configuration is found in the Flash memory.
Enable password:	Specify an enable password for this PIX Firewall.
Clock (UTC)	Set the PIX Firewall clock to Universal Coordinated Time (also known as Greenwich Mean Time).
Year [system year]:	Specify current year, or default to the year stored in the host computer.
Month [system month]:	Specify current month, or default to the month stored in the host computer.
Day [system day]:	Specify current day, or default to the day stored in the host computer.
Time [system time]	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 <i>inside</i> IP address. 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.

### Usage Guidelines

A PIX Firewall unit requires some initial configuration before PDM can connect to it. The setup dialog appears, via the console, at boot time if there is no configuration in the Flash memory. You can also access the **setup** command by typing **setup** from the Config mode.

The dialog asks for the inside IP address, network mask, host name, domain name and PDM host. The host and domain names are used to generate the default certificate for the SSL connection. The interface type is determined from 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 tftp flash](#)
- [http](#)

# session

Access an embedded AccessPro router console. (Privileged mode.)


**Note**

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

**session enable**

**no session**

**show session**


**Note**

Only use this command if you have an AccessPro router installed in your PIX Firewall.

**Syntax Description**

**enable** Enable the **session** command for communications with the AccessPro router.

**Usage Guidelines**

The **session** command allows you to specify Cisco IOS software commands on an AccessPro router console when the router is installed in your PIX Firewall. Use COM port 4 on the AccessPro router to communicate with the PIX Firewall.

Exit the router console session by entering tilde-dot (~.). Press the tilde key and when you hear a bell sound from your terminal, press the dot key.

While a router console session is occurring, the PIX Firewall disables failover because they both require the same interrupts.

**Examples**

This example enables an AccessPro session, starts the session, and then disables it:

```

session enable
Session has been enabled.
session

Warning: FAILOVER has been disabled!!!
Attempting session with embedded router, use ~. to quit!

acpro> ~.

no session
Session has been disabled
session
Session is not enabled

```

# show

View command information. (Differs by mode.)

**show ?**

## Usage Guidelines

The **show** command without arguments or the **show ?** command allows you to view the names of the **show** commands and their descriptions. Explanations for each **show** command are provided on the respective command page for the command itself where appropriate; for example, **show arp** is described on the **arp** command page.



### Note

The **show** commands that do not have a command equivalent shown in this section are described on their respective command pages; for example, the **show interface** command is described on the **interface** command page.

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

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

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

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

## Examples

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

```
show ?
?                help ...
```

# show blocks/clear blocks

Show system buffer utilization. (Privileged mode.)

**clear blocks**

**show blocks**

## Usage Guidelines

The **show blocks** command lists preallocated system buffer utilization. In the **show blocks** listing, the SIZE column displays the block type. The MAX column is the maximum number of allocated blocks. The LOW column is the fewest blocks available since last reboot. The CNT column is the current number of available blocks. A zero in the LOW column indicates a previous event where memory exhausted. A zero in the CNT column means memory is exhausted now. Exhausted memory is not a problem as long as traffic is moving through the PIX Firewall. You can use the **show conn** command to see if traffic is moving. If traffic is not moving and the memory is exhausted, a problem may be indicated.

The **clear blocks** command keeps the maximum count to whatever number is allocated in the system and equates the low count to the current count.

You can also view the information from the **show blocks** command using SNMP.

## 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 checksum
```

---

## Usage Guidelines

The **show checksum** command displays four groups of hexadecimal numbers that act as a digital summary of the contents of the configuration. This same information stores with the configuration when you store it in Flash memory. By using the **show config** command and viewing the checksum at the end of the configuration listing and using the **show checksum** command, you can compare the numbers to see if the configuration has changed. The PIX Firewall tests the checksum to determine if a configuration has not been corrupted.

---

## Examples

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

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

# show conn

Display all active connections. (Privileged mode.)

```
show conn [count] [foreign | local ip [-ip2]] [netmask mask] [protocol tcp | udp | protocol]
[fport | lport port1 [-port2]] [state [up [,finin] [,finout] [,http_get] [,sip] [,smtp_data]
[,smtp_banner] [,smtp_incomplete] [,nojava] [,data_in] [,data_out] [,sqlnet_fixup_data]
[,conn_inbound] [,rpc] [,h323] [,dump]]]
```

```
show conn [count] |

[protocol <tcp|udp>]

[foreign|local <ip1[-ip2]> [netmask <mask>]]

[lport|fport <port1[-port2]>]

[state <up[,finin][,finout][,http_get][,smtp_data]
[,nojava][,data_in][,data_out][,rpc][,h323]
[,sqlnet_fixup_data][,conn_inbound][,sip]>]
```

**pixfirewall(config)# show conn help**

```
usage: show conn [count] |

[protocol <tcp|udp>]

[foreign|local <ip1[-ip2]> [netmask <mask>]]

[lport|fport <port1[-port2]>]

[state <up[,finin][,finout][,http_get][,smtp_data]
[,nojava][,data_in][,data_out][,rpc][,h323]
[,sqlnet_fixup_data][,conn_inbound][,sip]>]
```

```
show conn state <up[,finin][,finout][,http_get][,smtp_data]
[,nojava][,data_in][,data_out][,rpc][,h323]
[,sqlnet_fixup_data][,conn_inbound][,sip]>]
```

## 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>foreign   local ip [-ip2]</b> <b>netmask mask</b>	Display active connections by the foreign IP address or by local IP address. Qualify foreign or local active connections by network mask.

<b>protocol</b> <i>tcp</i>   <i>udp</i>   <i>protocol</i>	Display active connections by protocol type. <i>protocol</i> is a protocol specified by number. See “Protocols” in Chapter 1, “Using PIX Firewall Commands” for a list of valid protocol literal names.
<b>fport</b>   <b>lport</b> <i>port1</i> [- <i>port2</i> ]	Display foreign or local active connections by port. See “Ports” in Chapter 1, “Using PIX Firewall Commands” for a list of valid port 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 and information about the active TCP connections. You can also view the connection count information from the **show conn** command using SNMP.

**Examples**

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	crypto engine	Displays usage statistics for the firewall cryptography engine.
--------------------	---------------	---

Command Modes	Privileged or configuration mode.
---------------	-----------------------------------

Usage Guidelines	The <b>show crypto engine</b> command displays usage statistics for the cryptography engine used by the firewall.
------------------	---

Examples	The following example shows sample output for the <b>show crypto engine</b> 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 lines. (Privileged mode.)

```
show history
```

## Usage Guidelines

The **show history** command displays previously entered commands. You can examine commands individually with the up and down arrows or by entering **^p** to view previously entered lines or **^n** to view the next line.

## Examples

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

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

# show interface

See the **interface** command page for a description of the **show interface** command.

# show memory

Show system memory utilization. (Privileged mode.)

**show memory**

---

## Usage Guidelines

The **show memory** command displays a summary of the maximum physical memory and current free memory available to the PIX Firewall operating system. Memory in the PIX Firewall is allocated as needed.

You can also view the information from the **show memory** command using SNMP.

---

## Examples

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

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

# show processes

Display processes. (Privileged mode.)

## show processes

### Usage Guidelines

The **show processes** command displays a listing of running processes. Processes are lightweight threads requiring only a few instructions. In the listing, PC is the program counter, SP is the stack pointer, STATE is the address of a thread queue, Runtime is the number of milliseconds that the thread has been running, SBASE is the stack base address, Stack is the current number of bytes used and the total size of the stack, and Process lists the thread's function.

### Examples

The following is sample output 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 tech-support

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

**show tech-support**

---

## Usage Guidelines

The **show tech-support** command lists information technical support analysts need to help you diagnose PIX Firewall problems. This command combines the output from the **show** commands that provide the most information to a technical support analyst.

---

## Examples

The following is sample output 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.)

**clear traffic**

**show traffic**

## Usage Guidelines

The **show traffic** command lists the number of packets and bytes moving through each interface. The number of seconds is the duration the PIX Firewall has been online since the last reboot. The **clear traffic** command clears counters for the **show traffic** command output.

## Examples

The following is sample output 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

See the **uauth** command page for information on the **show uauth** command.

# show version

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

**show version**

## Usage Guidelines

The **show version** command allows you to view the PIX Firewall unit's software version, operating time since last reboot, processor type, Flash memory type, interface boards, serial number (BIOS ID), and activation key value.

The uptime value in the output of 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.

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 serial number listed with the **show version** command, in version 5.3 and later, is for the Flash memory BIOS. This number is different from the serial number on the chassis. When you get a software upgrade, you will need the serial number that appears in the **show version** command, not the chassis number.

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

- Flash AT29C040A @ 0x300 for 2 MB of Flash
- Flash i28F640J5 @ 0x300 for 16 MB of Flash

## Examples

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

**show version**

```
Cisco Secure PIX Firewall Version 6.1(0)
Compiled on Fri 01-Oct-01 13:56 by pixbuild

pix515 up 4 days 22 hours 10 mins 42 secs

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

0: ethernet0: address is 00aa.0000.0037, irq 11
1: ethernet1: address is 00aa.0000.0038, irq 10
2: ethernet2: address is 00a0.c92a.f029, irq 9
3: ethernet3: address is 00a0.c948.45f9, irq 7

Licensed Features:
Failover:      Enabled
VPN-DES:      Enabled
VPN-3DES:     Disabled
Maximum Interfaces: 6

Serial Number: 123 (0x7b)
Activation Key: 0xc5233151 0xb429f6d0 0xda93739a 0xe15cdf51
```

## show xlate

See the **xlate** command page for information on the **show xlate** command.

## shun

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

```
[no] shun src_ip [dst_ip sport dport [protocol]]
clear shun [statistics]
show shun src_ip
```

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

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.

### 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.

### 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.

## snmp-server

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

```
snmp-server community key
snmp-server contact text
snmp-server location text
snmp-server host [if_name] ip_addr [trap | poll]
snmp-server enable traps
clear snmp-server command
no snmp-server command
show snmp-server
```

### 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.  The <i>key</i> is a case-sensitive value up to 32 characters in length. Spaces are not permitted. The default, if this option is not used, is <b>public</b> .
<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>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>snmp-server 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>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.
<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 five SNMP management stations. <p>Use with these parameters:</p> <ul style="list-style-type: none"> <li>• <i>if_name</i>—The interface name where the SNMP management station resides.</li> <li>• <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.</li> </ul>
<b>enable traps</b>	Enable or disable sending SNMP trap notifications via syslog.

### Usage Guidelines

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

In understanding SNMP use, the PIX Firewall is considered the SNMP agent or SNMP server. The management station is the system running the SNMP program that receives and processes the SNMP information that the PIX Firewall sends.

An SNMP object ID (OID) for PIX Firewall displays in SNMP event traps sent from the PIX Firewall. OID 1.3.6.1.4.1.9.1.227 was assigned as the PIX Firewall system object ID.

The **clear snmp-server** and **no snmp-server** commands removes command statements. The **show snmp-server** command displays the information.

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.

Accessibility to the PIX Firewall 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 and Cisco Memory Pool MIB are now available. These MIBs provide the following PIX Firewall information via SNMP:

- Buffer usage from the **show block** command
- Connection count from the **show conn** command
- Failover status

- Memory usage from the **show memory** command

### Receiving SNMP Requests from an SNMP Management Station

To receive SNMP requests from a management station:

- 
- Step 1** Identify the management station with an **snmp-server host** command statement.
- Step 2** Specify **snmp-server** command options for the **location**, **contact**, and **community**.
- Step 3** Start the SNMP software on the management station and begin issuing SNMP requests to the PIX Firewall.
- 

### Defaults

If you do not specify either 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 via Secure Shell (SSH). (Configuration mode.)

```
ssh disconnect session_id

no ssh disconnect session_id

ssh ip_address [netmask] [interface_name]

no ssh ip_address [netmask] [interface_name]

ssh timeout mm

no timeout mm

show ssh [sessions [ip_address]]

show ssh timeout

clear ssh
```

## Syntax Description

<i>ip_address</i>	IP address of the host or network authorized to initiate an SSH connection to the PIX Firewall.
<i>netmask</i>	Network mask for <i>ip_address</i> . If you do not specify a <i>netmask</i> , the default is 255.255.255.255 regardless of the class of <i>ip_address</i> .
<i>interface_name</i>	PIX Firewall interface name on which the host or network initiating the SSH connection resides.
<i>mm</i>	The duration in minutes that a session can be idle before being disconnected. The default duration is 5 minutes. The allowable range is from 1 to 60 minutes.
<i>session_id</i>	SSH session ID number available from the <b>show ssh sessions</b> command.

## Usage Guidelines

The **ssh *ip\_address*** command specifies the host or network authorized to initiate an SSH connection to the PIX Firewall. The **ssh timeout** command allows you to 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 allows you to disconnect a specific session you observed from the **show ssh sessions** command. Use the **clear ssh** command to remove all **ssh** command statements from the configuration. Use the **no ssh** command to remove selected **ssh** command statements from the configuration.



### Note

You must generate an RSA key-pair for the PIX Firewall before clients can connect to the PIX Firewall console. To use SSH, your PIX Firewall must have a DES or 3DES activation key.

To gain access to the PIX Firewall console via SSH, at the SSH client, enter the username as **pix** and enter the Telnet password. You can set the Telnet password with the **passwd** command; the default Telnet password is **cisco**. To authenticate using AAA server instead, configure the **aaa authenticate ssh console** command.

SSH permits up to 100 characters in a username and up to 50 characters in a password.

When starting an SSH session, a dot (.) displays on the PIX Firewall console before the SSH user authentication prompt appears.

The dot appears as follows:

```
pixfirewall(config)# .
pixfirewall(config)# .
```

The display of the dot does not affect the functionality of SSH. The dot appears on at the console when generating a server key or decrypting a message using private keys during SSH key exchange, before user authentication occurs. These tasks can take up to two minutes or longer. The dot is a progress indicator that verifies that the PIX Firewall is busy and has not hung.

### show ssh sessions Command

The **show ssh sessions** command provides the following display:

Session ID	Client IP	Version	Encryption	State	Username
0	172.16.25.15	1.5	3DES	4	-
1	172.16.38.112	1.5	DES	6	pix
2	172.16.25.11	1.5	3DES	4	-

The Session ID is a unique number that identifies an SSH session. The Client IP is the IP address of the system running an SSH client. The Version lists the 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 an **aaa authentication ssh console** *group\_tag* command statement is not defined, you can gain access to the PIX Firewall console with the username **pix** and with the PIX Firewall Telnet password (set with the **passwd** command). If the **aaa** command is defined, but the SSH authentication request times out, this implies that the AAA server may be down or not available. You can gain access to the PIX Firewall using the username **pix** and the enable password (set with the **enable password** command). By default, the Telnet password is **cisco** and the enable password is not set. If the enable password is empty (null), even if you enter the password correctly, you are not granted access to the SSH session.

The user authentication attempt limit is set to 3. Note that the Linux version of the SSH version 1 client available from <http://www.openssh.com> only allows one user authentication attempt.

---

**Examples**

Create an RSA key-pair with a modulus size of 1024 bits (recommended for use with Cisco IOS software):

```
hostname cisco-pix
domain-name example.com
ca generate rsa key 1024
show ca mypubkey rsa
ca save all
```

These command statements set the hostname and domain name for the PIX Firewall, generate the RSA key-pair, display the RSA key-pair, and save the RSA key-pair to Flash memory.

Start an SSH session so clients on the outside interface can access the PIX Firewall console remotely over a secure shell:

```
ssh 10.1.1.1 255.255.255.255 outside
ssh timeout 60
```

Configure the PIX Firewall to perform user authentication using AAA servers. The protocol is the protocol used by the AAA-server to do the authentication. The following example uses the TACACS+ authentication protocol.

```
aaa-server ssh123 (inside) host 10.1.1.200 mysecure
aaa-server ssh123 protocol tacacs+
aaa authenticate ssh console ssh123
```

---

**Related Commands**

- [aaa](#)
- [ca](#)
- [domain-name](#)
- [enable password](#)
- [hostname](#)
- [passwd](#)

# static

Maps a local IP address to a global IP address (NAT) and supports TCP and UDP port redirection (static PAT). (Configuration mode.)

```
[no] static [(internal_if_name, external_if_name)] {tcp | udp} {global_ip | interface} global_port
local_ip local_port [netmask mask] [max_conns [em_limit]] [norandomseq]
```

**show static**

## Syntax Description

<i>internal_if_name</i>	The internal network interface name. The higher security level interface you are accessing.
<i>external_if_name</i>	The external network interface name. The lower security level interface you are accessing.
<b>tcp</b>	Specifies TCP port redirection.
<b>udp</b>	Specifies UDP port redirection.
<b>interface</b>	The outside interface address is taken to be the global address.
<i>global_port</i>	Global TCP or UDP port for port redirection.
<i>local_port</i>	Local TCP or UDP port for port redirection.
<i>global_ip</i>	The global IP address used for redirection. The IP address on the lower security level interface you are accessing.
<i>local_ip</i>	The local IP address from the inside network. The IP address on the higher security level interface you are accessing.
<b>netmask</b>	Reserve word required before specifying the network mask.
<i>mask</i>	Pertains to both <i>global_ip</i> and <i>local_ip</i> . For host addresses, always use 255.255.255.255. For network addresses, use the appropriate class mask or subnet mask; for example, for Class A networks, use 255.0.0.0. An example subnet mask is 255.255.255.224.
<i>max_conns</i>	The maximum number of connections permitted through the static at the same time.
<i>em_limit</i>	The embryonic connection limit. An embryonic connection is one that has started but not yet completed. Set this limit to prevent attack by a flood of embryonic connections. The default is 0, which means unlimited connections.
<b>norandomseq</b>	Do not randomize the TCP/IP packet's sequence number. Only use this option if another inline firewall is also randomizing sequence numbers and the result is scrambling the data. Use of this option opens a security hole in the PIX Firewall.

## Usage Guidelines

The **static** command creates a permanent mapping (called a static translation slot or “xlate”) between a local IP address and a global IP address. Use the **static** and **access-list** commands when you are accessing an interface of a higher security level from an interface of a lower security level; for example, when accessing the inside from a perimeter or the outside interface.

### TCP Intercept Feature

Prior to version 5.3, 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 interface names on the **static** command may seem confusing at first. This is further complicated by how NAT is handled on the PIX Firewall. If NAT is disabled, with the **nat 0** command, statics are specified with a different set of rules than when NAT is enabled. For either no NAT or NAT, the rule of which command to access an interface stays the same as shown in [Table 8-1](#).

[Table 8-1](#) assumes that the security levels are 40 for dmz1 and 60 for dmz2.

**Table 8-1** 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>

**Table 8-1 Interface Access Commands by Interface (continued)**

From This Interface	To This Interface	Use This Command
outside	dmz2	<b>static</b>
outside	inside	<b>static</b>

**With NAT Enabled**

Network Address Translation (NAT) is enabled with the **nat** *n* command where “*n*” has the value **1** or greater; for example, **nat 1 0 0**.

Always specify the interface name of the highest security level interface you are accessing, followed by the lower security level interface. The IP addresses are also confusing because the first IP address you specify is for the lower security level interface. The second IP address is for the higher security level interface. The way to remember this is as follows.

**static** (*high,low*) *low high*

For example, assume you have four interfaces on the PIX Firewall that have security levels set with the **nameif** command as follows:

```
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 dmz1 security40
nameif ethernet3 dmz2 security60
```

To access the inside from the outside interface, you need a **static** command like the following:

```
static (inside,outside) outside_ip_address inside_ip_address netmask mask
```

Replace *outside\_ip\_address* with the global IP address (an IP address on the lower security level interface). Replace *inside\_ip\_address* with the IP address of the host on the higher security level interface that you want to grant access to.

Use these replacements in the rest of the commands in this section. Replace *mask* with 255.255.255.255 for host addresses, except when subnetting is in effect; for example, 255.255.255.128. For network addresses, use the appropriate class mask; for example, for Class A networks, use 255.0.0.0.

To access the inside from the dmz1 interface, you need a **static** command like the following:

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

To access the inside from the dmz2 interface, you need a **static** command like the following:

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

To access the dmz2 interface from the dmz1 interface, you need a **static** command like the following:

```
static (dmz2,dmz1) dmz1_ip_address dmz2_ip_address netmask mask
```

To go the other way around, from a higher security level interface to a lower security level interface, use the **nat** and **global** commands. For example, to access dmz1 from dmz2, use 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** mapping.

The first IP address you specify in the **static** command is the first IP address you specify in the **access-list** command as shown in this example:

```
static (dmz2,dmz1) 10.1.1.1 192.168.1.1 netmask 255.255.255.255
access-list acl_dmz1 permit tcp 10.1.1.0 255.255.255.0 host 10.1.1.1
access-group acl_dmz1 in interface dmz1
```

The **static** command maps the address 10.1.1.1 on the dmz1 interface so that users on the dmz1 interface can access the 192.168.1.1 host on the dmz2 interface. The **access-list** command lets any users in the 10.1.1.0 network access the 10.1.1.1 address over any TCP port. The **access-group** command statement binds the **access-list** command statement to the dmz1 interface.

**Note**

Always make **access-list** command statements as specific as possible. Using the **any** option to allow any host access should be used with caution for access lists used with statics.

**With No-NAT**

With no-NAT, the **static** command has a different sense of logic. With NAT disabled, addresses on both sides of the PIX Firewall are registered addresses. Between interfaces, addresses must be on different subnets that you control with subnetting. See “Appendix D” of 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.

See also: **access-list**

#### Examples

The example that follows creates a **static** command and then permits users to call in through H.323 using Intel InternetPhone, CU-SeeMe, CU-SeeMe Pro, MeetingPoint, or MS NetMeeting to 10.1.1.2 using IP address 209.165.201.2, to 10.1.1.10 using IP address 209.165.201.10, and so on. The net **static** command that follows maps addresses 209.165.201.1 through 209.165.201.30 to local addresses 10.1.1.1 through 10.1.1.30.

```
static (inside, outside) 209.165.201.0 10.1.1.0 netmask 255.255.255.255
access-list acl_out permit tcp any 209.165.201.0 255.255.255.224 eq h323
access-group acl_out in interface outside
```

The following example shows the commands used to disable Mail Guard:

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

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

# syslog

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


**Note**

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

# sysopt

Change PIX Firewall system options. (Configuration mode.)

```
sysopt connection permit-pptp
no sysopt connection permit-pptp
```

```
sysopt connection permit-l2tp
sysopt connection permit-ipsec
```

```
sysopt connection permit-ipsec
no sysopt connection permit-ipsec
sysopt connection permit-l2tp
no sysopt connection permit-l2tp
```

```
sysopt connection tcpmss bytes
no sysopt connection tcpmss bytes
```

```
sysopt connection timewait
no sysopt connection timewait
```

```
sysopt ipsec pl-compatible
no sysopt ipsec pl-compatible
```

```
sysopt nodnsalias inbound
sysopt nodnsalias outbound
no sysopt nodnsalias inbound
no sysopt nodnsalias outbound
```

```
sysopt noproxyarp if_name
no sysopt noproxyarp if_name
```

```
sysopt security fragguard
no sysopt security fragguard
```

```
sysopt radius ignore-secret
no sysopt radius ignore-secret
```

```
sysopt route dnat
no sysopt route dnat
```

```
sysopt uauth allow-http-cache
```

**no sysopt uauth allow-http-cache**

**clear sysopt**

**show sysopt**

### 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>route dnat</b>	Specify that when an incoming packet does a route lookup, the incoming interface is used to determine which interface the packet should go to, and which is the next hop.
<b>security fragguard</b>	Enable the IP Frag Guard feature.
<b>radius ignore-secret</b>	Ignore authenticator key to avoid retransmit caveat.
<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.

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.

#### **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.

### Examples

In the following example, a PPTP client authenticates using **mschap**, negotiates **mppe** encryption, receives the **dns** and **wins** server addresses, and Telnets to the host 192.168.0.2 directly through the **nat 0** command.

```
ip local pool my-addr-pool 10.1.1.1-10.1.1.254
aaa-server my-aaa-server-group (inside) host 192.168.0.10 key 12345678
aaa-server my-aaa-server-group protocol radius
vpdn group 1 accept dialin pptp
vpdn group 1 ppp authentication mschap
vpdn group 1 ppp encryption mppe auto required
vpdn group 1 client configuration address local my-addr-pool
vpdn group 1 client authentication aaa my-aaa-server-group
vpdn group 1 client configuration dns 10.2.2.99
vpdn group 1 client configuration wins 10.2.2.100
vpdn enable outside
access-list nonat permit ip 10.1.1.0 255.255.255.0 host 192.168.0.2
access-list nonat permit ip 10.1.1.0 255.255.255.0 host 10.2.2.99
access-list nonat permit ip 10.1.1.0 255.255.255.0 host 10.2.2.100
nat (inside) 0 access-list nonat
sysopt connection permit-pptp
```

#### **sysopt connection 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
```

### **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.

$$1380 \text{ data} + 20 \text{ TCP} + 20 \text{ IP} + 24 \text{ AH} + 24 \text{ ESP\_CIPHER} + 12 \text{ ESP\_AUTH} + 20 \text{ IP} = 1500 \text{ bytes}$$

1500 bytes is the MTU for Ethernet connections. We recommend that the default value of 1380 bytes be used for Ethernet and mixed Ethernet and Token Ring environments. If the PIX Firewall has all Token Ring interfaces, you can set *bytes* to 4056. However, if even one link along the path through the network is not a Token Ring, setting *bytes* to such a high value may cause poor throughput. In its 1380 byte default value, this command increases throughput of the **sysopt security fragguard** command.

The TCP maximum segment size is the maximum size that an end host can inject into the network at one time (see RFC 793 for more information on the TCP protocol). The **sysopt connection tcpmss** command is recommended in a network environment being attacked being with overly aggressive TCP or HTTP stack with a faulty path MTU value that is degrading the performance of the PIX Firewall IP Frag Guard feature. Environments where one or more end hosts reside on a Token Ring network are especially susceptible to this faulty behavior.

**Note**

Although, not advised for normal use of this feature, if you encounter the syslog IPFRAG messages 209001 and 209002, you can raise the *bytes* value.

**sysopt connection timewait**

The **sysopt connection timewait** command is necessary for end host applications whose default TCP terminating sequence is a simultaneous close instead of the normal shutdown sequence (see RFC 793). In a simultaneous close, both ends of the transaction initiate the closing sequence, as opposed to the normal sequence where one end closes and the other end acknowledges prior to initiating its own closing sequence.

The default behavior of the PIX Firewall is to track the normal shutdown sequence and release the connection after two FINs and the ACKnowledgment of the last FIN segment. This quick release heuristic enables the PIX Firewall to sustain a high connection rate.

However with a simultaneous close, the quick release forces one side of the connection to linger in the CLOSING state (see RFC 793). Many sockets in the CLOSING state can degrade the performance of an end host. For instance, some WinSock mainframe clients are known to exhibit this behavior and degrade the performance of the mainframe server. Old versions of HP/UX are also susceptible to this behavior. Enabling the **sysopt connection timewait** command enables a quiet time window for the abnormal close down sequence to complete.

The **no sysopt connection timewait** command disables the option, which is off by default.



#### Note

Use of the **sysopt connection timewait** command may impact PIX Firewall performance especially with low memory configuration and highly dynamic traffic pattern such as HTTP.

#### **sysopt nodnsalias**

The **sysopt nodnsalias inbound** disables inbound embedded DNS A record fixups according to aliases that apply to the A record address. **sysopt nodnsalias outbound** affects outbound replies.

This command remedies the case when a DNS server is on the outside and users on the inside need to access a server on a perimeter interface. In the past, you would use the **alias** command to permit DNS responses to resolve correctly through the PIX Firewall, but formerly you had to reverse the parameters for the local IP address and foreign IP address.

For example, you would normally code the **alias** command as follows:

```
alias (inside) 192.168.1.4 209.165.201.11 255.255.255.255
```

Inside host 192.168.1.5 needs access to www.example.com, which resolves at an outside ISP DNS to 209.165.201.11. The PIX Firewall fixes this DNS response sending the host a response of 192.168.1.4. The host uses its gateway (the PIX Firewall) to go to 192.168.1.4, which the PIX Firewall now aliases back to the 209.165.201.11. Because this is actually 192.168.1.4, a server on the perimeter interface of the PIX Firewall, the packet is dropped because the PIX Firewall sent the packet to the outside interface, which is the incorrect interface.

The **sysopt nodnsalias inbound** command has the same effect as reversing the **alias** command statement parameters as follows:

```
alias (inside) 209.165.201.11 192.168.1.4 255.255.255.255
```

This works properly because everything happens in reverse. The DNS is now modified to 209.165.201.11 and the host inside uses its gateway (the PIX Firewall) to get there, the PIX Firewall aliases this back to 192.168.1.4 and routes it out the perimeter interface to the correct host and the TCP connection is established.

#### **sysopt noproxyarp**

The **sysopt noproxyarp** command allows you to disable proxy-arps on a PIX Firewall interface.

#### **sysopt radius ignore-secret**

Some commonly used RADIUS servers, such as Livingston version 1.16, have a usage caveat where they do not include the key in the authenticator hash in the accounting acknowledgment response. This can cause the PIX Firewall to continually retransmit the accounting request. Use the **sysopt radius**

**ignore-secret** command to cause the PIX Firewall to ignore the key in the authenticator of accounting acknowledgments thus avoiding the retransmit problem. (The key described here is the key you set with the **aaa-server** command.)

#### **sysopt route dnat**

The **sysopt route dnat** command specifies that when an incoming packet does a route lookup, the incoming interface is used to determine which interface the packet should go to, and which is the next hop.

#### **sysopt security fragguard**

The **sysopt security fragguard** command enables the IP Frag Guard feature. This feature is disabled by default. This feature enforces two additional security checks in addition to the security checks recommended by RFC 1858 against the many IP fragment style attacks: teardrop, land, and so on. First, each non-initial IP fragment is required to be associated with an already seen valid initial IP fragment. Second, IP fragments are limited to 100 full IP fragmented packets per second to each internal host.

The IP Frag Guard feature operates on all interfaces in the PIX Firewall and cannot be selectively enabled or disabled by interface.

PIX Firewall uses the **security fragguard** command to enforce the security policy determined by a **access-list permit** or **access-list deny** command to permit or deny packets through the PIX Firewall.



#### **Note**

Use of the **sysopt security fragguard** command breaks normal IP fragmentation conventions. However, not using this command exposes PIX Firewall to the possibility of IP fragmentation attacks. We recommend that packet fragmentation not be permitted on the network if at all possible.

If PIX Firewall is used as a tunnel for FDDI packets between routers, disable the **security fragguard** command feature.

Because Linux sends IP fragments in reverse order, fragmented Linux packets will not pass through the PIX Firewall with the **sysopt security fragguard** command enabled.

The **show sysopt** command lists the **sysopt** commands in the configuration. The **clear sysopt** command resets the **sysopt** command to default settings. The **no sysopt security fragguard** command disables the IP Frag Guard feature.

#### **Examples**

The following example disables IP Frag Guard and then lists the current command options:

```
no sysopt security fragguard
show sysopt
sysopt security fragguard
no sysopt connection tcpmss
no sysopt connection timewait
```