



## PIX Firewall System Management

---

This chapter describes how to configure and use the tools and features provided by the PIX Firewall for monitoring and configuring the system, and for monitoring network activity. It contains the following sections:

- [Using Telnet for Remote System Management](#)
- [IDS Syslog Messages](#)
- [Using DHCP](#)
- [Using SNMP](#)
- [Using SSH](#)

### Using Telnet for Remote System Management



**Note**

Access to the console via Telnet is available on the inside and third interfaces. The third interface is the network connecting to the third usable slot in the PIX Firewall. You can view the third interface with the **show nameif** command. The third entry from the top of the listing is the third interface.

The serial console lets a single user configure the PIX Firewall, but many times this is not convenient for a site with more than one administrator. PIX Firewall allows you to access the serial console via Telnet from hosts on any internal interface. With IPSec configured, you can use Telnet to remotely administer the console of a PIX Firewall from the outside interface. This section contains the following sections:

- [Configuring Telnet Console Access](#)
- [Testing Telnet Access](#)
- [Securing a Telnet Connection on the Outside Interface](#)
- [Trace Channel Feature](#)

## Configuring Telnet Console Access

Follow these steps to configure Telnet console access:

- Step 1** Use the PIX Firewall **telnet** command. For example, to let a host on the internal interface with an address of 192.168.1.2 access the PIX Firewall, enter the following.

```
telnet 192.168.1.2 255.255.255.255 inside
```

If IPSec is in place, you can let a host on the outside interface access the PIX Firewall console. Refer to [“Securing a Telnet Connection on the Outside Interface”](#) for more information. Use a command such as the following.

```
telnet 209.165.200.225 255.255.255.224 outside
```

- Step 2** If required, set the duration for how long a Telnet session can be idle before PIX Firewall disconnects the session. The default duration, 5 minutes, is too short in most cases and should be increased until all pre-production testing and troubleshooting has been completed. Set a longer idle time duration as shown in the following example.

```
telnet timeout 15
```

- Step 3** If you want to protect access to the console with an authentication server, you can use the **aaa authentication telnet console** command, which requires that you have a username and password on the authentication server. When you access the console, PIX Firewall prompts you for these login credentials. If the authentication server is offline, you can still access the console by using the username **pix** and the password set with the **enable password** command.

- Step 4** Save the commands in the configuration using the **write memory** command.

## Testing Telnet Access

Perform the following steps to test Telnet access:

- Step 1** From the host, start a Telnet session to a PIX Firewall interface IP address. If you are using Windows 95 or Windows NT, click **Start>Run** to start a Telnet session. For example, if the inside interface IP address is 192.168.1.1, enter the following command.

```
telnet 192.168.1.1
```

- Step 2** The PIX Firewall prompts you with a password:

```
PIX passwd:
```

Enter **cisco** and press the **Enter** key. You are then logged into the PIX Firewall.

The default password is **cisco**, which you can change with the **passwd** command.

You can enter any command on the Telnet console that you can set from the serial console, but if you reboot the PIX Firewall, you will need to log back into the PIX Firewall after it restarts.

Some Telnet applications such as the Windows 95 or Windows NT Telnet sessions may not support access to the PIX Firewall’s command history feature used with the arrow keys. However, you can access the last entered commands by pressing Ctrl-P.

- Step 3** Once you have Telnet access available, you may want to view ping information while debugging. You can view ping information from Telnet sessions with the **debug icmp trace** command. The Trace Channel feature also affects **debug** displays, which is explained in “[Trace Channel Feature](#).”

Messages from a successful ping appear as follows:

```
Outbound ICMP echo request (len 32 id 1 seq 512) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 256) 209.165.201.1 > 209.165.201.23
```

- Step 4** In addition, you can use the Telnet console session to view syslog messages:
- Start message displays with the **logging monitor 7** command. The “7” will cause all syslog message levels to display.  
  
If you are using the PIX Firewall in production mode, you may wish to use the **logging buffered 7** command to store messages in a buffer that you can view with the **show logging** command, and clear the buffer for easier viewing with the **clear logging** command. To stop buffering messages, use the **no logging buffered** command.  
  
You can also lower the number from 7 to a lesser value, such as 3, to limit the number of messages that appear.
  - If you entered the **logging monitor** command, then enter the **terminal monitor** command to cause the messages to display in your Telnet session. To disable message displays, use the **terminal no monitor** command.

---

[Example 7-1](#) shows commands for using Telnet to permit host access to the PIX Firewall console.

**Example 7-1 Using Telnet**

```
telnet 10.1.1.11 255.255.255.255
telnet 192.168.3.0 255.255.255.0
```

The first **telnet** command permits a single host, 10.1.1.11 to access the PIX Firewall console with Telnet. The 255 value in the last octet of the netmask means that only the specified host can access the console.

The second **telnet** command permits PIX Firewall console access from all hosts on the 192.168.3.0 network. The 0 value in the last octet of the netmask permits all hosts in that network access. However, Telnet only permits 16 hosts simultaneous access to the PIX Firewall console over Telnet.

## Securing a Telnet Connection on the Outside Interface

This section tells you how to secure your PIX Firewall console Telnet connection to the outside interface of the PIX Firewall. It includes the following topics:

- [Overview](#)
- [Using Cisco Secure VPN Client](#)
- [Using Cisco VPN 3000 Client](#)

## Overview

If you are using the Cisco Secure Policy Manager, version 2.0 or later, this section also applies to you. It is assumed you are using the Cisco VPN Client version 3.0, Cisco Secure VPN Client version 1.1, or the Cisco VPN 3000 Client version 2.5, to secure your Telnet connection. In the example in the next section, the IP address of the PIX Firewall's outside interface is 168.20.1.5, and the Cisco Secure VPN Client's IP address stemming from the virtual pool of addresses is 10.1.2.0.

See the **telnet** command page within the *Cisco PIX Firewall Command Reference* for more information about this command.



**Note** You will need to have two security policies set up on your VPN client. One security policy is used to secure your Telnet connection and another to secure your connection to the inside network.

## Using Cisco Secure VPN Client

This section applies only if you are using a Cisco Secure VPN Client. To encrypt your Telnet connection to the PIX Firewall's outside interface, perform the following steps as part of your PIX Firewall configuration.

- 
- Step 1** Create an **access-list** command statement to define the traffic to protect from the PIX Firewall to the VPN client using a destination address from the virtual local pool of addresses:
- ```
access-list 80 permit ip host 168.20.1.5 10.1.2.0 255.255.255.0
```
- Step 2** Specify which host can access the PIX Firewall console with Telnet:
- ```
telnet 10.1.2.0 255.255.255.0 outside
```
- Specify the VPN client's address from the local pool and the outside interface.
- Step 3** Within the VPN client, create a security policy that specifies the Remote Party Identity IP address and gateway IP address as the same IP address—the IP address of the PIX Firewall's outside interface. In this example, the IP address of the PIX Firewall's outside is 168.20.1.5.
- Step 4** Configure the rest of the security policy on the VPN client to match the PIX Firewall's security policy.
- 

## Using Cisco VPN 3000 Client

This section applies only if you are using a Cisco VPN 3000 Client. To encrypt your Telnet connection to the PIX Firewall's outside interface, perform the following step as part of your PIX Firewall configuration. In the following example, the IP address of the PIX Firewall's outside interface is 168.20.1.5, and the Cisco VPN 3000 Client's IP address stemming from the virtual pool of addresses is 10.1.2.0.

Specify which host can access the PIX Firewall console with Telnet. Specify the VPN client's address from the local pool and the outside interface.

```
telnet 10.1.2.0 255.255.255.0 outside
```

## Trace Channel Feature

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

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

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

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

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



### Note

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

## IDS Syslog Messages

PIX Firewall lists single-packet (*atomic*) Cisco Intrusion Detection System (IDS) signature messages via syslog. Refer to *Cisco PIX Firewall System Log Messages* for a list of the supported messages. You can view this document online at the following website:

[http://www.cisco.com/en/US/products/sw/secursw/ps2120/products\\_system\\_message\\_guide\\_book09186a008008d2bc.html](http://www.cisco.com/en/US/products/sw/secursw/ps2120/products_system_message_guide_book09186a008008d2bc.html)

All signature messages are not supported by PIX Firewall in this release. IDS syslog messages all start with `%PIX-4-4000nn` and have the following format:

```
%PIX-4-4000nn IDS:sig_num sig_msg from ip_addr to ip_addr on interface int_name
```

For example:

```
%PIX-4-400013 IDS:2003 ICMP redirect from 10.4.1.2 to 10.2.1.1 on interface dmz
%PIX-4-400032 IDS:4051 UDP Snork attack from 10.1.1.1 to 192.168.1.1 on interface outside
```



### Note

Cisco IDS signature number 1101 is not supported by PIX Firewall. When an unsupported signature number is entered, PIX Firewall returns an error message.

Options:

- sig\_num** The signature number. Refer to the *Cisco Secure Intrusion Detection System Version 2.2.1 User Guide* for more information. You can view the “NSDB and Signatures” chapter from this guide at the following website:  
[http://www.cisco.com/en/US/partner/products/sw/secursw/ps2308/products\\_user\\_guide\\_list.html](http://www.cisco.com/en/US/partner/products/sw/secursw/ps2308/products_user_guide_list.html)
- sig\_msg** The signature message—approximately the same as the NetRanger signature message.
- ip\_addr** The local to remote address to which the signature applies.
- int\_name** The name of the interface on which the signature originated.

You can determine which messages display with the following commands:

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

Attaches a global policy to a signature. Used to disable or exclude a signature from auditing.

**no ip audit signature *signature\_number***

Removes the policy from a signature. Used to reenable a signature.

**show ip audit signature [*signature\_number*]**

Displays disabled signatures.

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

Specifies the default action to be taken for signatures classified as informational signatures.

The **alarm** option indicates that when a signature match is detected in a packet, PIX Firewall reports the event to all configured syslog servers. The **drop** option drops the offending packet. The **reset** option drops the offending packet and closes the connection if it is part of an active connection. The default is **alarm**. To cancel event reactions, specify the **ip audit info** command without an **action** option.

**no ip audit info**

Sets the action to be taken for signatures classified as informational and reconnaissance to the default action.

**show ip audit info**

Displays the default informational actions.

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

Specifies the default actions to be taken for attack signatures. The **action** options are as previously described.

**no ip audit attack**

Sets the action to be taken for attack signatures to the default action.

### **show ip audit attack**

Displays the default attack actions. An audit policy (audit rule) defines the attributes for all signatures that can be applied to an interface along with a set of actions. Using an audit policy the user may limit the traffic that is audited or specify actions to be taken when the signature matches. Each audit policy is identified by a name and can be defined for informational or attack signatures. Each interface can have two policies; one for informational signatures and one for attack signatures. If a policy is defined without actions, then the configured default actions will take effect. Each policy requires a different name.

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

All informational signatures except those disabled or excluded by the **ip audit signature** command are considered part of the policy. The actions are the same as described previously.

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

Remove the audit policy *audit\_name*.

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

All attack signatures except those disabled or excluded by the **ip audit signature** command are considered part of the policy. The actions are the same as described previously.

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

Removes the audit specification *audit\_name*.

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

Displays all audit policies or specific policies referenced by name and possibly type.

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

Applies an audit specification or policy (via the ip audit name command) to an interface.

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

Removes a policy from an interface.

### **show ip audit interface**

Displays the interface configuration.

## Using DHCP

PIX Firewall supports Dynamic Host Configuration Protocol (DHCP) servers and DHCP clients. DHCP is a protocol that supplies automatic configuration parameters to Internet hosts. This protocol has two components:

- Protocol for delivering host-specific configuration parameters from a DHCP server to a host (DHCP client)
- Mechanism for allocating network addresses to hosts

A DHCP server is simply a computer that provides configuration parameters to a DHCP client, and a DHCP client is a computer or network device that uses DHCP to obtain network configuration parameters.

The primary purpose of implementing the DHCP server and DHCP client features into the PIX Firewall is to significantly simplify the configuration of a PIX Firewall unit.

This section includes the following topics:

- [DHCP Client](#)
- [DHCP Server](#)

## DHCP Client

DHCP client support within the PIX Firewall is designed for use within a small office, home office (SOHO) environment using a PIX Firewall that is directly connected to a DSL or cable modem that supports the DHCP server function. With the DHCP client feature enabled on a PIX Firewall, the PIX Firewall functions as a DHCP client to a DHCP server allowing the server to configure the unit's enabled interface with an IP address, subnet mask, and optionally a default route.



### Note

Use of the DHCP client feature to acquire an IP address from a generic DHCP server is not supported. Also, the PIX Firewall DHCP client does not support **failover** configurations.

To support the DHCP client feature within the PIX Firewall, the following enhancements were made:

- Enhanced the **ip address** and the **show ip address** commands:
  - **ip address if\_name dhcp** [setroute] [retry **retry\_cnt**]
  - **ip address outside dhcp** [setroute] [retry *retry\_cnt*]
  - **show ip address if\_name dhcp**
- Added new **debug** commands:
  - **debug dhcpc packet**
  - **debug dhcpc detail**
  - **debug dhcpc error**

The **ip address dhcp** command enables the DHCP client feature on the specified PIX Firewall interface. The optional **setroute** argument tells the PIX Firewall to set the default route using the default gateway parameter the DHCP server returns.

The **debug dhcpc** commands provide debugging tools for the enabled DHCP client feature.

The PIX Firewall commands used to implement the DHCP client are described in the **ip address** command page and the **debug** command page in the [Cisco PIX Firewall Command Reference](#). Refer to these command pages for more information.



### Note

The DHCP-acquired IP address of the outside interface can also be used as the PAT global address. This makes it unnecessary for the ISP to assign a static IP address to PIX Firewall. Use the **global** command with **interface** keyword to enable PAT to use the DHCP-acquired IP address of outside interface. For more information about the **global** command, see the **global** command page in the [Cisco PIX Firewall Command Reference](#).

## Enabling the DHCP Client Feature and Setting Default Route

To enable the DHCP client feature on a given PIX Firewall interface and set the default route via the DHCP server, configure the **ip address dhcp setroute** command as part of your entire PIX Firewall configuration, including the **setroute** option. Specify the name of the interface on which the DHCP client will be enabled.

## DHCP Server

DHCP server support within the PIX Firewall is designed for use within a remote home or branch office (ROBO) environment using a PIX 506 unit. Connecting to the PIX Firewall are PC clients and other network devices (DHCP clients) that establish network connections that are either insecure (unencrypted) or secure (encrypted using IPSec) to access an enterprise or corporate network. As a DHCP server, the PIX Firewall provides network configuration parameters to the DHCP clients through the use of DHCP. These configuration parameters provide a DHCP client the networking parameters used to access the enterprise network, and once in the network, the network services to use, such as the DNS server.

Prior to the version 5.3 software release, PIX Firewall DHCP servers supported 10 DHCP clients. PIX Firewall version 5.3 and later supports 32 DHCP clients on PIX Firewall and 256 on other platforms. In version 6.0 or later, PIX Firewall DHCP server supports up to 256 DHCP clients. You cannot configure a DHCP server for 256 clients, using a Class C netmask. For example, if a company has a Class C network address of 172.17.1.0 with netmask 255.255.255.0, then 172.17.1.0 (network IP) and 172.17.1.255 (broadcast) cannot be in the DHCP address pool range. Further, one address is used up for the PIX Firewall interface. Thus, if a user uses a Class C netmask, they can only have up to 253 DHCP Clients. To have 256 clients configured, they cannot use a Class C netmask.

**Note**

---

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

---

The PIX Firewall commands used to implement the DHCP server feature are described in the **dhcpcd** command page and the **debug** command page in the *Cisco PIX Firewall Command Reference*. Refer to these command pages for more information.

## Configuring the DHCP Server Feature

Be sure to configure the IP address and the subnet mask of the **inside** interface using the **ip address** command prior to enabling the DHCP server feature.

Follow these steps to enable the DHCP server feature on a given PIX Firewall interface. (Steps 1 and 6 are required.)

- 
- Step 1** Specify a DHCP address pool using the **dhcpcd address** command. The PIX Firewall will assign to a client one of the addresses from this pool to use for a given length of time. The default is the **inside** interface.

For example:

```
dhcpcd address 10.0.1.101-10.0.1.110 inside
```

- Step 2** (Optional) Specify the IP address(es) of the DNS server(s) the client will use. You can specify up to two DNS servers. For example:

```
dhcpcd dns 209.165.201.2 209.165.202.129
```

- Step 3** (Optional) Specify the IP address(es) of the WINS server(s) the client will use. You can specify up to two WINS servers.
- For example:
- ```
dhcpd wins 209.165.201.5
```
- Step 4** Specify the lease length to grant the client. This lease equals the amount of time (in seconds) the client can use its allocated IP address before the lease expires. The default value is 3600 seconds.
- For example:
- ```
dhcpd lease 3000
```
- Step 5** (Optional) Configure the domain name the client will use.
- For example:
- ```
dhcpd domain example.com
```
- Step 6** Enable the DHCP daemon within the PIX Firewall to listen for DHCP client requests on the enabled interface. Currently, you can only enable the DHCP server feature on the **inside** interface, which is the default.
- For example:
- ```
dhcpd enable inside
```

---

The following example shows a configuration listing for the previous procedure.

```
! set the ip address of the inside interface
ip address inside 10.0.1.2 255.255.255.0
! configure the network parameters the client will use once in the corporate network and
dhcpd address 10.0.1.101-10.0.1.110
dhcpd dns 209.165.201.2 209.165.202.129
dhcpd wins 209.165.201.5
dhcpd lease 3000
dhcpd domain example.com
! enable dhcp server daemon on the inside interface
dhcpd enable inside
```

The following example shows the configuration of a DHCP address pool and a DNS server address with the inside interface being enabled for the DHCP server feature:

```
dhcpd address 10.0.1.100-10.0.1.108
dhcpd dns 209.165.200.227
dhcpd enable
```

The following example shows the configuration of a DHCP address pool and uses the **auto\_config** command to configure the dns, wins, and domain parameters:

```
dhcpd address 10.0.1.100-10.0.1.108
dhcpd auto_config
dhcpd enable
```

The following is a partial configuration example of the DHCP server and IPSec features configured on a PIX Firewall that is within a remote office. The PIX 506 unit's VPN peer is another PIX Firewall that has an outside interface IP address of 209.165.200.228 and functions as a gateway for a corporate network.

```
! configure interface ip address
ip address outside 209.165.202.129 255.255.255.0
ip address inside 172.17.1.1 255.255.255.0
```

```
! configure ipsec with corporate pix
access-list ipsec-peer permit ip 172.17.1.0 255.255.255.0 192.168.0.0 255.255.255.0
ipsec transform-set myset esp-des esp-sha-hmac
crypto map mymap 10 ipsec-isakmp
crypto map mymap 10 match address ipsec-peer
crypto map mymap 10 set transform-set myset
crypto map mymap 10 set peer 209.165.200.228
crypto map mymap interface outside
sysopt connection permit-ipsec
nat (inside) 0 access-list ipsec-peer
isakmp policy 10 authentication preshare
isakmp policy 10 encryption des
isakmp policy 10 hash sha
isakmp policy 10 group 1
isakmp policy 10 lifetime 3600
isakmp key 12345678 address 0.0.0.0 netmask 0.0.0.0
isakmp enable outside
!configure dhcp server address
dhcpd address 172.17.1.100-172.17.1.109
dhcpd dns 192.168.0.20
dhcpd wins 192.168.0.10
dhcpd lease 3000
dhcpd domain example.com
! enable dhcp server on inside interface
dhcpd enable
! use outside interface ip as PAT global address
nat (inside) 1 0 0
global (outside) 1 interface
```

## Using SNMP

The `snmp-server` command causes the PIX Firewall to send SNMP traps so that the PIX Firewall can be monitored remotely. Use `snmp-server host` command to specify which systems receive the SNMP traps.

This section includes the following topics:

- [Introduction](#)
- [MIB Support](#)
- [SNMP Usage Notes](#)
- [SNMP Traps](#)
- [Compiling Cisco Syslog MIB Files](#)
- [Using the Firewall and Memory Pool MIBs](#)

## Introduction

The PIX Firewall SNMP MIB-II groups available are System and Interfaces. The Cisco Firewall MIB and Cisco Memory Pool MIB are also available.

All SNMP values are read only (RO).

Using SNMP, you can monitor system events on the PIX Firewall. SNMP events can be read, but information on the PIX Firewall cannot be changed with SNMP.

The PIX Firewall SNMP traps available to an SNMP management station are as follows:

- Generic traps:
  - Link up and link down (cable connected to the interface or not; cable connected to an interface working or not working)
  - Cold start
  - Authentication failure (mismatched community string)
- Security-related events sent via the Cisco Syslog MIB:
  - Global access denied
  - Failover syslog messages
  - syslog messages

Use CiscoWorks for Windows or any other SNMP V1, MIB-II compliant browser to receive SNMP traps and browse an MIB. SNMP traps occur at UDP port 162.

## MIB Support



### Note

---

The PIX Firewall does not support browsing of the Cisco syslog MIB.

---

You can browse the System and Interface groups of MIB-II. Browsing an MIB is different from sending traps. Browsing means doing an **snmpget** or **snmpwalk** of the MIB tree from the management station to determine values.

## MIB Support

The Cisco Firewall MIB and Cisco Memory Pool MIB are available.

PIX Firewall does not support the following in the Cisco Firewall MIB:

- cfwSecurityNotification NOTIFICATION-TYPE
- cfwContentInspectNotification NOTIFICATION-TYPE
- cfwConnNotification NOTIFICATION-TYPE
- cfwAccessNotification NOTIFICATION-TYPE
- cfwAuthNotification NOTIFICATION-TYPE
- cfwGenericNotification NOTIFICATION-TYPE

## SNMP Usage Notes

- The MIB-II ifEntry.ifAdminStatus object returns 1 if the interface is accessible and 2 if you administratively shut down the interface using the **shutdown** option of the **interface** command.
- The SNMP “ifOutUcastPkts” object now correctly returns the outbound packet count.
- Syslog messages generated by the SNMP module now specify the interface name instead of an interface number.

## SNMP Traps

Traps are different than browsing; they are unsolicited “comments” from the managed device to the management station for certain events, such as link up, link down, and syslog event generated.

An SNMP object ID (OID) for PIX Firewall displays in SNMP event traps sent from the PIX Firewall. PIX Firewall provides system OID in SNMP event traps & SNMP mib-2.system.sysObjectID variable based on the hardware platform:

Table 7-1 lists the system OID in PIX Firewall platforms:

**Table 7-1 System OID in PIX Firewall Platforms**

PIX Platform	System OID
PIX 506	.1.3.6.1.4.1.9.1.389
PIX 515	.1.3.6.1.4.1.9.1.390
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 (original PIX Firewall OID)

Two mechanisms work with SNMP, PIX Firewall responds to an SNMP request from a management station and the PIX Firewall sends a trap, which is an event notification. PIX Firewall supports two types of traps, generic and syslog traps.

## Receiving Requests and Sending Syslog Traps

Follow these steps to receive requests and send traps from the PIX Firewall to an SNMP management station:

- 
- Step 1** Identify the IP address of the SNMP management station with the **snmp-server host** command.
- Step 2** Set the **snmp-server** options for **location**, **contact**, and the **community** password as required.
- If you only want to send the cold start, link up, and link down generic traps, no further configuration is required.
- If you only want to receive SNMP requests, no further configuration is required.
- Step 3** Add an **snmp-server enable traps** command statement.
- Step 4** Set the logging level with the **logging history** command:
- ```
logging history debugging
```
- We recommend that you use the **debugging** level during initial set up and during testing. Thereafter, set the level from **debugging** to a lower value for production use.
- (The **logging history** command sets the severity level for SNMP syslog messages.)
- Step 5** Start sending syslog traps to the management station with the **logging on** command.
- Step 6** To disable sending syslog traps, use the **no logging on** command or the **no snmp-server enable traps** command.
-

The commands in [Example 7-2](#) specify that PIX Firewall can receive the SNMP requests from host 192.168.3.2 on the inside interface but does not send SNMP syslog traps to any host.

**Example 7-2 Enabling SNMP**

```
snmp-server host 192.168.3.2
snmp-server location building 42
snmp-server contact polly hedra
snmp-server community ohwhatakeyisthee
```

The **location** and **contact** commands identify where the host is and who administers it. The **community** command specifies the password in use at the PIX Firewall SNMP agent and the SNMP management station for verifying network access between the two systems.

## Compiling Cisco Syslog MIB Files

To receive security and failover SNMP traps from the PIX Firewall, compile the Cisco SMI MIB and the Cisco syslog MIB into your SNMP management application. If you do not compile the Cisco syslog MIB into your application, you only receive traps for link up or down, firewall cold start and authentication failure.

You can select Cisco MIB files for PIX Firewall and other Cisco products from the following website:

- <http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>

From this page, select **PIX Firewall** from the Cisco Secure & VPN selection list.

Follow these steps to compile Cisco syslog MIB files into your browser using CiscoWorks for Windows (SNMPc):

- 
- Step 1** Get the Cisco syslog MIB files.
  - Step 2** Start SNMPc.
  - Step 3** Click **Config>Compile MIB**.
  - Step 4** Scroll to the bottom of the list, and click the last entry.
  - Step 5** Click **Add**.
  - Step 6** Find the Cisco syslog MIB files.




---

**Note** With certain applications, only files with a .mib extension may show in the file selection window of the SNMPc. The Cisco syslog MIB files with the .my extension will not be shown. In this case, you should manually change the .my extension to a .mib extension.

---

- Step 7** Click CISCO-FIREWALL-MIB.my (CISCO-FIREWALL-MIB.mib) and click **OK**.
- Step 8** Scroll to the bottom of the list, and click the last entry.
- Step 9** Click **Add**.
- Step 10** Find the file CISCO-MEMORY-POOL-MIB.my (CISCO-MEMORY-POOL-MIB.mib) and click **OK**.
- Step 11** Scroll to the bottom of the list, and click the last entry.
- Step 12** Click **Add**.
- Step 13** Find the file CISCO-SMI.my (CISCO-SMI.mib) and click **OK**.

- Step 14** Scroll to the bottom of the list, and click the last entry.
- Step 15** Click **Add**.
- Step 16** Find the file CISCO-SYSLOG-MIB.my (CISCO-SYSLOG-MIB.mib) and click **OK**.
- Step 17** Click **Load All**.
- Step 18** If there are no errors, restart SNMPc.



---

**Note** These instructions are only for SNMPc (CiscoWorks for Windows).

---

## Using the Firewall and Memory Pool MIBs

The Cisco Firewall and Memory Pool MIBs let you poll failover and system status.

This section contains the following topics:

- [ipAddrTable Notes](#)
- [Viewing Failover Status](#)
- [Verifying Memory Usage](#)
- [Viewing The Connection Count](#)
- [Viewing System Buffer Usage](#)

In the tables that follow in each section, the meaning of each returned value is shown in parentheses.

### ipAddrTable Notes

- Use of the SNMP ip.ipAddrTable entry requires that all interfaces have unique addresses. If interfaces have not been assigned IP addresses, by default, their IP addresses are all set to 127.0.0.1. Having duplicate IP addresses causes the SNMP management station to loop indefinitely. The workaround is to assign each interface a different address. For example, you can set one address to 127.0.0.1, another to 127.0.0.2, and so on.

SNMP uses a sequence of GetNext operations to traverse the MIB tree. Each GetNext request is based on the result of the previous request. Therefore, if two consecutive interfaces have the same IP 127.0.0.1 (table index), the GetNext function returns 127.0.0.1, which is correct; however, when SNMP generates the next GetNext request using the same result (127.0.0.1), the request is identical to the previous one, which causes the management station to loop infinitely.

For example:

```
GetNext(ip.ipAddrTable.ipAddrEntry.ipAdEntAddr.127.0.0.1)
```

In SNMP protocol, the MIB table index should be unique for the agent to identify a row from the MIB table. The table index for ip.ipAddrTable is the PIX Firewall interface IP address, so the IP address should be unique; otherwise, the SNMP agent will get confused and may return information of another interface (row), which has the same IP (index).

## Viewing Failover Status

The Cisco Firewall MIB's `cfsHardwareStatusTable` allows you to determine whether failover is enabled and which unit is active. The Cisco Firewall MIB indicates failover status by two rows in the `cfwHardwareStatusTable` object. From the PIX Firewall command line, you can view failover status with the **show failover** command. You can access the object table from the following path:

```
.iso.org.dod.internet.private.enterprises.cisco.ciscoMgmt.ciscoFirewallMIB.
ciscoFirewallMIBObjects.cfwSystem.cfwStatus.cfwHardwareStatusTable
```

Table 7-2 lists which objects provide failover information.

**Table 7-2 Failover Status Objects**

| Object                                        | Object Type                  | Row 1: Returned if Failover is Disabled | Row 1: Returned if Failover is Enabled                             | Row 2: Returned if Failover is Enabled                             |
|-----------------------------------------------|------------------------------|-----------------------------------------|--------------------------------------------------------------------|--------------------------------------------------------------------|
| <code>cfwHardwareType</code><br>(table index) | Hardware                     | 6 (If primary unit)                     | 6 (If primary unit)                                                | 7 (If secondary unit)                                              |
| <code>cfwHardwareInformation</code>           | <code>SnmpAdminString</code> | blank                                   | blank                                                              | blank                                                              |
| <code>cfwHardwareStatusValue</code>           | HardwareStatus               | 0 (Not used)                            | active or 9 (If active unit)<br>or standby or 10 (If standby unit) | active or 9 (If active unit)<br>or standby or 10 (If standby unit) |
| <code>cfwHardwareStatusDetail</code>          | <code>SnmpAdminString</code> | <b>Failover Off</b>                     | blank                                                              | blank                                                              |

In the HP OpenView Browse MIB application's "MIB values" window, if failover is disabled, a sample MIB query yields the following information:

```
cfwHardwareInformation.6 :
cfwHardwareInformation.7 :
cfwHardwareStatusValue.6 :0
cfwHardwareStatusValue.7 :0
cfwHardwareStatusDetail.6 :Failover Off
cfwHardwareStatusDetail.7 :Failover Off
```

From this listing, the table index, `cfwHardwareType`, appears as either `.6` or `.7` appended to the end of each of the subsequent objects. The `cfwHardwareInformation` field is blank, the `cfwHardwareStatusValue` is **0**, and the `cfwHardwareStatusDetail` contains **Failover Off**, which indicates the failover status.

When failover is enabled, a sample MIB query yields the following information:

```
cfwHardwareInformation.6 :
cfwHardwareInformation.7 :
cfwHardwareStatusValue.6 : active
cfwHardwareStatusValue.7 : standby
cfwHardwareStatusDetail.6 :
cfwHardwareStatusDetail.7 :
```

In this listing, only the `cfwHardwareStatusValue` contains values, either **active** or **standby** to indicate the status of each unit.

## Verifying Memory Usage

You can determine how much free memory is available with the Cisco Memory Pool MIB. From the PIX Firewall command line, memory usage is viewed with the **show memory** command. The following is sample output from the **show memory** command.

```
show memory
16777216 bytes total, 5595136 bytes free
```

You can access the MIB objects from the following path:

```
.iso.org.dod.internet.private.enterprises.cisco.ciscoMgmt.ciscoMemoryPoolMIB.
ciscoMemoryPoolObjects.ciscoMemoryPoolTable
```

Table 7-3 lists which objects provide memory usage information.

**Table 7-3** Memory Usage Objects

| Object                               | Object Type          | Returned Value                                                                         |
|--------------------------------------|----------------------|----------------------------------------------------------------------------------------|
| ciscoMemoryPoolType<br>(table index) | CiscoMemoryPoolTypes | 1 (Processor memory)                                                                   |
| ciscoMemoryPoolName                  | DisplayString        | <b>PIX system memory</b>                                                               |
| ciscoMemoryPoolAlternate             | Integer32            | 0 (No alternate memory pool)                                                           |
| ciscoMemoryPoolValid                 | TruthValue           | <b>true</b> (Means that the values of the remaining objects are valid)                 |
| ciscoMemoryPoolUsed                  | Gauge32              | <i>integer</i> (Number of bytes currently in use—the total bytes minus the free bytes) |
| ciscoMemoryPoolFree                  | Gauge32              | <i>integer</i> (Number of bytes currently free)                                        |
| ciscoMemoryPoolLargestFree           | Gauge32              | 0 (Information not available)                                                          |

In the HP OpenView Browse MIB application's "MIB values" window a sample MIB query yields the following information:

```
ciscoMemoryPoolName.1 :PIX system memory
ciscoMemoryPoolAlternate.1 :0
ciscoMemoryPoolValid.1 :true
ciscoMemoryPoolUsed.1 :12312576
ciscoMemoryPoolFree.1 :54796288
ciscoMemoryPoolLargestFree.1 :0
```

From this listing, the table index, ciscoMemoryPoolName, appears as the .1 value at the end of each subsequent object value. The ciscoMemoryPoolUsed object lists the number of bytes currently in use, **12312576**, and the ciscoMemoryPoolFree object lists the number of bytes currently free **54796288**. The other objects always list the values described in Table 7-3.

## Viewing The Connection Count

You can view the number of connections in use from the `cfwConnectionStatTable` in the Cisco Firewall MIB. From the PIX Firewall command line, you can view the connection count with the **show conn** command. The following is sample output from the **show conn** command to demonstrate where the information in `cfwConnectionStatTable` originates.

```
show conn
15 in use, 88 most used
```

The `cfwConnectionStatTable` object table can be accessed from the following path:

```
.iso.org.dod.internet.private.enterprises.cisco.ciscoMgmt.ciscoFirewallMIB.
ciscoFirewallMIBObjects.cfwSystem.cfwStatistics.cfwConnectionStatTable
```

Table 7-4 lists which objects provide connection count information.

**Table 7-4 Connection Count Objects**

| Object                                                 | Object Type     | Row 1: Returned Value                                                | Row 2: Returned Value                                                            |
|--------------------------------------------------------|-----------------|----------------------------------------------------------------------|----------------------------------------------------------------------------------|
| <code>cfwConnectionStatService</code><br>(Table index) | Services        | <b>40</b> (IP protocol)                                              | <b>40</b> (IP protocol)                                                          |
| <code>cfwConnectionStatType</code><br>(Table index)    | ConnectionStat  | <b>6</b> (Current connections in use)                                | <b>7</b> (High)                                                                  |
| <code>cfwConnectionStatDescription</code>              | SnmpAdminString | <b>number of connections currently in use by the entire firewall</b> | <b>highest number of connections in use at any one time since system startup</b> |
| <code>cfwConnectionStatCount</code>                    | Counter32       | <b>0</b> (Not used)                                                  | <b>0</b> (Not used)                                                              |
| <code>cfwConnectionStatValue</code>                    | Gauge32         | <i>integer</i> (In use number)                                       | <i>integer</i> (Most used number)                                                |

In the HP OpenView Browse MIB application's "MIB values" window a sample MIB query yields the following information:

```
cfwConnectionStatDescription.40.6 :number of connections currently in use by the entire firewall
cfwConnectionStatDescription.40.7 :highest number of connections in use at any one time since system startup
cfwConnectionStatCount.40.6 :0
cfwConnectionStatCount.40.7 :0
cfwConnectionStatValue.40.6 :15
cfwConnectionStatValue.40.7 :88
```

From this listing, the table index, `cfwConnectionStatService`, appears as the **.40** appended to each subsequent object and the table index, `cfwConnectionStatType`, appears as either **.6** to indicate the number of connections in use or **.7** to indicate the most used number of connections. The `cfwConnectionStatValue` object then lists the connection count. The `cfwConnectionStatCount` object always returns **0** (zero).

## Viewing System Buffer Usage

You can view the system buffer usage from the Cisco Firewall MIB in multiple rows of the `cfwBufferStatsTable`. The system buffer usage provides an early warning of the PIX Firewall reaching the limit of its capacity. On the command line, you can view this information with the **show blocks** command. The following is sample output from the **show blocks** command to demonstrate how `cfwBufferStatsTable` is populated.

```
show blocks
SIZE      MAX      LOW      CNT
   4      1600    1600    1600
   80      100     97      97
  256      80      79      79
 1550     780    402    404
65536      8       8       8
```

You can view `cfwBufferStatsTable` at the following path:

```
.iso.org.dod.internet.private.enterprises.cisco.ciscoMgmt.ciscoFirewallMIB.
ciscoFirewallMIBObjects.cfwSystem.cfwStatistics.cfwBufferStatsTable
```

Table 7-5 lists the objects required to view the system block usage.

**Table 7-5 System Block Usage Objects**

| Object                                          | Object Type        | First Row: Returned Value                                                                                         | Next Row: Returned Value                                                                                                    | Next Row: Returned Value                                                                                          |
|-------------------------------------------------|--------------------|-------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| <code>cfwBufferStatSize</code><br>(Table index) | Unsigned32         | <i>integer</i> (SIZE value; for example, 4 for a 4-byte block)                                                    | <i>integer</i> (SIZE value; for example, 4 for a 4-byte block)                                                              | <i>integer</i> (SIZE value; for example, 4 for a 4-byte block)                                                    |
| <code>cfwBufferStatType</code><br>(Table index) | ResourceStatistics | 3 (MAX)                                                                                                           | 5 (LOW)                                                                                                                     | 8 (CNT)                                                                                                           |
| <code>cfwBufferStatInformation</code>           | SnmpAdminString    | <b>maximum number of allocated <i>integer</i> byte blocks</b> ( <i>integer</i> is the number of bytes in a block) | <b>fewest <i>integer</i> byte blocks available since system startup</b> ( <i>integer</i> is the number of bytes in a block) | <b>current number of available <i>integer</i> byte blocks</b> ( <i>integer</i> is the number of bytes in a block) |
| <code>cfwBufferStatValue</code>                 | Gauge32            | <i>integer</i> (MAX number)                                                                                       | <i>integer</i> (LOW number)                                                                                                 | <i>integer</i> (CNT number)                                                                                       |



**Note**

The three rows repeat for every block size listed in the output of the **show blocks** command.

In the HP OpenView Browse MIB application's "MIB values" window a sample MIB query yields the following information:

```
cfwBufferStatInformation.4.3 :maximum number of allocated 4 byte blocks
cfwBufferStatInformation.4.5 :fewest 4 byte blocks available since system startup
cfwBufferStatInformation.4.8 :current number of available 4 byte blocks
cfwBufferStatInformation.80.3 :maximum number of allocated 80 byte blocks
cfwBufferStatInformation.80.5 :fewest 80 byte blocks available since system startup
cfwBufferStatInformation.80.8 :current number of available 80 byte blocks
cfwBufferStatInformation.256.3 :maximum number of allocated 256 byte blocks
cfwBufferStatInformation.256.5 :fewest 256 byte blocks available since system startup
cfwBufferStatInformation.256.8 :current number of available 256 byte blocks
cfwBufferStatInformation.1550.3 :maximum number of allocated 1550 byte blocks
```

```

cfwBufferStatInformation.1550.5 :fewest 1550 byte blocks available since system startup
cfwBufferStatInformation.1550.8 :current number of available 1550 byte blocks
cfwBufferStatValue.4.3: 1600
cfwBufferStatValue.4.5: 1600
cfwBufferStatValue.4.8: 1600
cfwBufferStatValue.80.3: 400
cfwBufferStatValue.80.5: 396
cfwBufferStatValue.80.8: 400
cfwBufferStatValue.256.3: 1000
cfwBufferStatValue.256.5: 997
cfwBufferStatValue.256.8: 999
cfwBufferStatValue.1550.3: 1444
cfwBufferStatValue.1550.5: 928
cfwBufferStatValue.1550.8: 932

```

From this listing, the first table index, `cfwBufferStatSize`, appears as first number appended to the end of each object, such as `.4` or `.256`. The other table index, `cfwBufferStatType`, appears as `.3`, `.5`, or `.8` after the first index. For each block size, the `cfwBufferStatInformation` object identifies the type of value and the `cfwBufferStatValue` object identifies the number of bytes for each value.

## Using SSH

This section describes how to use Secure Shell (SSH) to remotely manage a PIX Firewall in a secure way. It includes the following topics:

- [Overview](#)
- [Enabling SSH on the PIX Firewall](#)
- [Using an SSH Client](#)
- [Obtaining an SSH Client](#)

## Overview

SSH (Secure Shell) is an application running on top of a reliable transport layer, such as TCP/IP that provides strong authentication and encryption capabilities. PIX Firewall supports the SSH remote shell functionality as provided in SSH version 1. SSH version 1 also works with Cisco IOS software devices. Up to five SSH clients are allowed simultaneous access to the PIX Firewall console.

## Enabling SSH on the PIX Firewall



### 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 specify a host for PIX Firewall console access through Secure Shell (SSH), enter the following command:

```
[no] ssh ip_address [netmask] [interface_name]
```

Replace *ip\_address* with the IP address of the host or network authorized to initiate an SSH connection to the PIX Firewall. Replace *netmask* with the network mask for *ip\_address*. If you do not specify a *netmask*, the default is 255.255.255.255 regardless of the class of *ip\_address*. Replace *interface\_name* with name of the PIX Firewall interface on which the host or network initiating the SSH connection resides.

To limit the length of the SSH session, enter the following command:

```
ssh timeout mm
```

Replace *mm* with 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.

To disconnect a session, enter the following command:

```
ssh disconnect session_id
```

Replace *session\_id* with the SSH session ID number, which you can determine by entering the following command.

```
show ssh [sessions [ip_address]]
```

## Using an SSH Client

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.

To configure local authentication for an SSH client accessing the PIX Firewall from a Linux or UNIX command line, enter the following command:

```
ssh -c 3des -l pix -v ipaddress
```

Use the **-c** option to identify the cipher used. PIX Firewall accepts **3des** and **des**. Use the **-l** option to identify the password used for connecting to the PIX Firewall. If no authentication is enabled on the SSH connection, use the default user name **pix**. Use the **-v** option to enable verbose mode, and replace *ipaddress* with the address of the PIX Firewall.



### Note

Windows and Macintosh SSH clients typically have graphic interfaces where you enter the required information.

The password used to perform local authentication is the same as the one used for Telnet access. The default for this password is **cisco**. To change this password, enter the following command:

```
passwd string
```

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

## 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/>