G through L Commands

global

Create or delete entries from a pool of global addresses.

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

clear global

show global

Syntax Description

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

Command Modes

Configuration mode.

Usage Guidelines

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

When used on a PPPoE interface, the **global** command should explicitly include a netmask. Otherwise, the 255.255.255.255 netmask, assigned to the interface by PPPoE, is used as the broadcast mask. In that case, all addresses in the global pool may become broadcast addresses and will become unusable for address translation.

Use caution with names that contain a "-" (dash) character because the **global** command interprets the last (or only) "-" character in the name as a range specifier instead of as part of the name. For example, the **global** command treats the name "`host-net2`" as a range from "`host`" to "`net2`". If the name is "`host-net2-section3`" then it is interpreted as a range from "`host-net2`" to "`section3`".

The following command form is used for Port Address Translation (PAT) only:
global [(*if_name*)] *nat_id* {{*global_ip*} [**netmask** *global_mask*] | **interface**}

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

Use the **no global** command to remove access to a *nat_id*, or to a Port Address Translation (PAT) address, or address range within a *nat_id*.

The **show global** command displays the **global** command statements in the configuration.

PAT

You can enable the Port Address Translation (PAT) feature by entering a single IP address with the **global** command. PAT lets multiple outbound sessions appear to originate from a single IP address. With PAT enabled, the PIX Firewall chooses a unique port number from the PAT IP address for each outbound xlate (translation slot). This feature is valuable when an Internet service provider cannot allocate enough unique IP addresses for your outbound connections. An IP address you specify for a PAT cannot be used in another global address pool.

When a PAT augments a pool of global addresses, first the addresses from the global pool are used, then the next connection is taken from the PAT address. If a global pool address is available, the next connection takes that address. The global pool addresses always come first, before a PAT address is used. Augment a pool of global addresses with a PAT by using the same *nat_id* in the **global** command statements that create the global pools and the PAT.

For example:

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

PAT does not work with H.323 applications and caching nameservers. Do not use a PAT when multimedia applications need to be run through the PIX Firewall. Multimedia applications can conflict with port mappings provided by PAT.

The firewall does not PAT all ICMP message types; it only PATs ICMP echo and echo-reply packets (types 8 and 0). Specifically, only ICMP echo or echo-reply packets create a PAT xlate. So, when the other ICMP messages types are dropped, syslog message 305006 (on the PIX Firewall) is generated.

PAT does not work with the **established** command. PAT works with DNS, FTP and passive FTP, HTTP, email, RPC, rshell, Telnet, URL filtering, and outbound traceroute.

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

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

```
global (outside) 1 209.165.201.5 netmask 255.255.255.224
```

To specify PAT using the IP address of an interface, specify the **interface** keyword in the **global** [(*int_name*)] *nat_id address* | **interface** command.

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

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

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

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

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

The following example maps hosts on the internal network 10.1.0.0/24 to global address 192.168.1.1 and hosts on the internal network 10.1.1.1/24 to global address 209.165.200.225 in global configuration mode.

```
nat (inside) 1 10.1.0.0 255.255.255.0
nat (inside) 2 10.1.1.0 255.255.255.0
global (outside) 1 192.168.1.1 netmask 255.255.255.0
global (outside) 2 209.165.200.225 netmask 255.255.255.224
```

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

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

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

PAT and DNS

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

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

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

For example, PAT is enabled with these commands:

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

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

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

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

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

Examples

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

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

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

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

help

Display help information.

help *command*

?

Syntax Description

?	Displays all commands available in the current privilege level and mode.
command	Specifies the PIX Firewall command for which to display the PIX Firewall command-line interface (CLI) help.
help	If no command name is specified, displays all commands available in the current privilege level and mode; otherwise, displays the PIX Firewall CLI help for the command specified.

Command Modes

Unprivileged, privileged, and configuration modes.

Usage Guidelines

The **help** or **?** command displays help information about all commands. You can view help for an individual command by entering the command name followed by a "**?**"(question mark).

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

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

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

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

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

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

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

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

hostname

Change the host name in the PIX Firewall command-line prompt.

hostname *newname*

Syntax Description

newname	Specifies a new host name for the firewall and is displayed in the firewall prompt. This name can be up to 63 characters, including alphanumeric characters, spaces or any of the following special characters: ‘( ) + - , . / : = ?

Command Modes Configuration mode.

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

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

Examples The following example shows how to change a host name:

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

http

Enables the PIX Firewall HTTP server and specifies the clients that are permitted to access it. Additionally, for access, the Cisco PIX Device Manager (PDM) requires that the PIX Firewall have an enabled HTTP server.

[**no**] **http** *ip_address* [*netmask*] [*if_name*]

[**no**] **http server enable**

clear http

show http

Syntax Description

clear http	Removes all HTTP hosts and disables the server.
http	Relating to the Hypertext Transfer Protocol.
http server enable	Enables the HTTP server required to run PDM.
if_name	PIX Firewall interface name on which the host or network initiating the HTTP connection resides.
ip_address	Specifies the host or network authorized to initiate an HTTP connection to the PIX Firewall.
netmask	Specifies the network mask for the **http** *ip_address*.

Defaults

If you do not specify a netmask, the default is **255.255.255.255** regardless of the class of IP address. The default *if_name* is **inside**.

Command Modes

Configuration mode.

Usage Guidelines

Access from any host will be allowed if **0.0.0.0 0.0.0.0** (or **0 0**) is specified for *ip_address* and *netmask.*

The **show http** command displays the allowed hosts and whether or not the HTTP server is enabled.

Examples

The following **http** command example is used for one host:

```
http 16.152.1.11 255.255.255.255 outside
```

The following **http** command example is used for any host:

```
http 0.0.0.0 0.0.0.0 inside
```

icmp

Configure access rules for Internet Control Message Protocol (ICMP) traffic that terminates at an interface.

[**no**] **icmp** {**permit** | **deny**} *ip_address net_mask* [*icmp_type*] *if_name*

clear icmp

show icmp

Syntax Description

deny	Deny access if the conditions are matched.
icmp_type	ICMP message type as described in Table 6-1.
if_name	The interface name.
ip_address	The IP address of the host sending ICMP messages to the interface.
net_mask	The mask to be applied to *ip_address*.
permit	Permit access if the conditions are matched.

Command Modes

Configuration mode.

Usage Guidelines

By default, the PIX Firewall denies all inbound traffic through the outside interface. Based on your network security policy, you should consider configuring the PIX Firewall to deny all ICMP traffic at the outside interface, or any other interface you deem necessary, by using the **icmp** command.

The **icmp** command controls ICMP traffic that received by the firewall. If no ICMP control list is configured, then the PIX Firewall accepts all ICMP traffic that terminates at any interface (including the outside interface), except that the PIX Firewall does not respond to ICMP echo requests directed to a broadcast address.

The **icmp deny** command disables pinging to an interface, and the **icmp permit** command enables pinging to an interface. With pinging disabled, the PIX Firewall cannot be detected on the network. This is also referred to as configurable proxy pinging.

For traffic that is routed through the PIX Firewall only, you can use the **access-list** or **access-group** commands to control the ICMP traffic routed through the PIX Firewall.

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

If an ICMP control list is configured, then the PIX Firewall uses a first match to the ICMP traffic followed by an implicit deny all. That is, if the first matched entry is a permit entry, the ICMP packet continues to be processed. If the first matched entry is a deny entry or an entry is not matched, PIX Firewall discards the ICMP packet and generates the %PIX-3-313001 syslog message. An exception is when an ICMP control list is not configured; in that case, a permit is assumed.

The syslog message is as follows:

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

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

ICMP Message Types

Table 6-1 lists possible ICMP type values.

Table 6-1 ICMP Type Literals

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

Examples

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

```
icmp permit any unreachable outside
```

The default behavior of the PIX Firewall is to deny ICMP messages to the outside interface.

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

```
icmp permit host 172.16.2.15 echo-reply outside
icmp permit 172.22.1.0 255.255.0.0 echo-reply outside
icmp permit any unreachable outside
```

igmp

Refer to the **multicast** command for the **igmp** subcommands.

The Internet Group Management Protocol (IGMP) enables IP hosts to report their multicast group memberships to an adjacent multicast router. On the PIX Firewall, IGMP support is implemented as a subcommand to the **multicast** command.

interface

Sets network interface parameters and configures VLANs.

interface *hardware_id* [*hardware_speed* [**shutdown**]]

[**no**] **interface** *hardware_id vlan_id* [**logical | physical**] [**shutdown**]

interface *hardware_id* **change-vlan** *old_vlan_id new_vlan_id*

clear interface

show interface *hardware_id* [*hardware_speed*] [**shutdown**]

Syntax Description

change-vlan	Keyword to change the VLAN identifier for an interface.
hardware_id	Identifies the network interface type. Possible values are **ethernet0**, **ethernet1** to **ethernet***n*, or **gb-ethernet***n,* depending on how many network interfaces are in the PIX Firewall.
hardware_speed	Network interface speed (optional). **aui**—Set 10 for Mbps Ethernet half-duplex communication with an AUI cable interface. **auto**—Negotiates Ethernet speed and duplex settings automatically. The **auto** keyword can only be used with the Intel 10/100 automatic speed-sensing network interface card. **bnc**—Set for 10 Mbps Ethernet half-duplex communication with a BNC cable interface. Possible Ethernet values are: **10baseT**—To set for 10 Mbps Ethernet half-duplex communication. **10full**—To set for 10 Mbps Ethernet full-duplex communication. **100baseTX**—To set for 100 Mbps Ethernet half-duplex communication. **100full**—To set for 100 Mbps Ethernet full-duplex communication. Possible Gigabit Ethernet (gb-ethernetX) values are: **1000auto**—To auto negotiate speed and duplex. **1000full**—To auto negotiate, advertising 1000 Mbps full duplex. **1000full nonegotiate**—To force link to 1000 Mbps full duplex.
logical	Creates a logical interface and applies the VLAN.
new_vlan_id	The new VLAN indentifier.
old_vlan_id	The current VLAN indentifier.
physical	Apply VLAN to physical interface.
shutdown	Disable an interface.
vlan_id	The VLAN identifier. For example: vlan10, vlan20, and so on.

Command Modes

Configuration mode.

Defaults

When configured, VLAN logical interfaces are enabled by default.

Usage Guidelines

The **interface** command sets the speed and duplex settings of the network interface boards, and brings up the interfaces specified. After changing an **interface** command, use the **clear xlate** command.

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

The i82542 Gigabit Ethernet interface currently used in the PIX Firewall does not support half duplex; as a result, **1000auto** is equivalent to **1000full** when using this interface.

VLAN interfaces

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

Physical interfaces are one per each NIC, in place at boot time, and non-removable. Logical interfaces that can be many-to-one for each NIC, are created at run time, and can be removed through software reconfiguration. A minimum of two physical interfaces are required for all PIX Firewall platforms to support VLANs.

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

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

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

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

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

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

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

Use the **show interface** command to display information about the VLAN configuration.

Use the **interface** *hardware_id vlan_id* **logical shutdown** command to temporarily disable a logical interface.

Use the **interface** *hardware_id* **change-vlan** *old_vlan_id new_vlan_id* command to reassign a VLAN.

Use the **no interface** *hardware_id vlan_id* **logical** command to remove the VLAN configuration.

no and clear commands

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

Use the **no interface** command to remove logical interfaces and VLAN definitions. (However, a **no interface** command does not negate an interface **shutdown** command.)

Note Using a **no interface** command on a logical interface (used for VLAN configuration) removes the logical interface from the system. Removing the logical interface also deletes all configuration rules applied to that interface, so exercise caution when using **no interface** commands with logical interfaces.

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

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

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

Note Even though the default is to set automatic speed sensing for the interfaces with the **interface** *hardware_id* **auto** command, we recommend that you specify the speed of the network interfaces; for example, **10baseT** or **100baseTX**. This lets PIX Firewall operate in network environments that may include switches or other devices that do not handle auto sensing correctly.

show interface

The **show interface** command lets you view network interface information for Ethernet. This is one of the first commands you should use when establishing network connectivity after installing a PIX Firewall.

Note The PIX 501 switch interface always indicates `100000 Kbit full duplex` (100,000 Kbps full duplex) even though the switch ports have negotiated the speed and duplex settings. The PIX Firewall automatically negotiates the inside interface setting at **100full** and this is not configurable.

Gigabit interface cards do not provide information for the extended **show interface** command counters introduced in Version 5.0(3). For Gigabit Ethernet interfaces, the current and maximum count for the number of blocks on the input (receive) queue will always be the same (63).

The information in the **show interface** command is as follows in Table 6-3:

Table 6-3 show interface Description

Show Interface Command Output	Description
Ethernet string	Indicates that you have used the **interface** command to configure the interface. The statement indicates either outside or inside, and whether the interface is available (“up”) or not available (“down”).
line protocol up or line protocol down	The message “line protocol up” means a working cable is plugged into the network interface. If the message is “line protocol down,” either the cable is incorrect or not plugged into the interface connector. The **show interface** command reports “line protocol down” for BNC cable connections and for 3Com cards.
Network interface type	Indicates type of network interface.
Interrupt vector	Note: It is acceptable for interface cards to have the same interrupts.
MAC address	Intel cards start with “i” and 3Com cards with “3c.”
Maximum transmission unit (MTU)	The size, in bytes, that data can best be sent over the network.
nn packets input	Indicates that packets are being received in the PIX Firewall.
nn packets output	Indicates that packets are being sent from the PIX Firewall.
Line duplex status	Half duplex indicates that the network interface switches back and forth between sending and receiving information; full duplex indicates that the network interface can send or receive information simultaneously.
Line speed	**10baseT** is listed as 10,000 Kb; **100baseTX** is listed as 100,000 Kb.

- The **show interface** command includes eight status counters (valid only for Ethernet interfaces).

  The following example shows sample output:

```
show interface
interface ethernet0 "outside" is up, line protocol is up
  Hardware is i82559 ethernet, address is 00aa.0000.003b
  IP address 209.165.201.7, subnet mask 255.255.255.224
  MTU 1500 bytes, BW 100000 Kbit half duplex
        1184342 packets input, 1222298001 bytes, 0 no buffer
        Received 26 broadcasts, 27 runts, 0 giants
        4 input errors, 0 CRC, 4 frame, 0 overrun, 0 ignored, 0 abort
        1310091 packets output, 547097270 bytes, 0 Andorrans, 0 unicast
           repave drops
        0 output errors, 28075 collisions, 0 interface resets
        0 babbles, 0 late collisions, 117573 deferred
        0 lost carrier, 0 no carrier
input queue (cure/max blocks): hardware (128/128) software (0/1)
        output queue (cure/max blocks): hardware (0/2) software (0/1)
```

The **show interface** counter descriptions are as follows in Table 6-4:

Table 6-4 show interface Counters

Counter	Description
output errors	The number of frames not transmitted because the configured maximum number of collisions was exceeded. This counter should only increment during heavy network traffic.
collisions	The number of messages retransmitted due to an Ethernet collision (single and multiple collisions). This usually occurs on an overextended LAN (Ethernet or transceiver cable too long, more than two repeaters between stations, or too many cascaded multiport transceivers). A packet that collides is counted only once by the output packets.
interface resets	The number of times an interface has been reset. If an interface is unable to transmit for three seconds, PIX Firewall resets the interface to restart transmission. During this interval, connection state is maintained. An interface reset can also happen when an interface is looped back or shut down.
babbles	Unused. (“babble” means that the transmitter has been on the interface longer than the time taken to transmit the largest frame.)
late collisions	The number of frames that were not transmitted because a collision occurred outside the normal collision window. A late collision is a collision that is detected late in the transmission of the packet. Normally, these should never happen. When two Ethernet hosts try to talk at once, they should collide early in the packet and both back off, or the second host should see that the first one is talking and wait. If you get a late collision, a device is jumping in and trying to send the packet on the Ethernet while the PIX Firewall is partly finished sending the packet. The PIX Firewall does not resend the packet, because it may have freed the buffers that held the first part of the packet. This is not a real problem because networking protocols are designed to cope with collisions by resending packets. However, late collisions indicate a problem exists in your network. Common problems are large repeated networks and Ethernet networks running beyond the specification.

Table 6-4 show interface Counters (continued)

Counter	Description
deferred	The number of frames that were deferred before transmission due to activity on the link.
lost carrier	The number of times the carrier signal was lost during transmission.
no carrier	Unused.
input queue (curr/max blocks)	Input queue—The input (receive) hardware and software queue. • hardware—(current and maximum blocks). The number of blocks currently present on the input hardware queue, and the maximum number of blocks previously present on that queue. In the example, there are currently 128 blocks on the input hardware queue, and the maximum number of blocks ever present on this queue was 128. • software—(current and maximum blocks). The number of blocks currently present on the input software queue, and the maximum number of blocks previously present on that queue. In the example, there are currently 0 blocks on the input software queue, and the maximum number of blocks ever present on this queue was 1.
output queue (curr/max blocks)	Output queue—The output (transmit) hardware and software queue. • hardware—(current and maximum blocks). The number of blocks currently present on the output hardware queue, and the maximum number of blocks previously present on that queue. In the example, there are currently 0 blocks on the output hardware queue, and the maximum number of blocks ever present on this queue was 2. • software—(current and maximum blocks). The number of blocks currently present on the output software queue, and the maximum number of blocks previously present on that queue. In the example, there are currently 0 blocks on the output software queue, and the maximum number of blocks ever present on this queue was 1.

Examples

The following example shows interface activity on the interface ethernet0, which has been named **outside**:

```
show interface
interface ethernet0 "outside" is up, line protocol is up
  Hardware is i82559 ethernet, address is 0000.0001.0001
  IP address 209.165.201.17, subnet mask 255.255.255.0
  MTU 1500 bytes, BW 10000 Kbit full duplex
        4203 packets input, 376390 bytes, 0 no buffer
        Received 3894 broadcasts, 0 runts, 0 giants
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
        1320 packets output, 123652 bytes, 0 underruns
        0 output errors, 0 collisions, 0 interface resets
        0 babbles, 0 late collisions, 0 deferred
        0 lost carrier, 0 no carrier
        input queue (curr/max blocks): hardware (35/128) software (0/2)
        output queue (curr/max blocks): hardware (0/1) software (0/1)
```

The following example sets a Gigabit Ethernet interface named **gb0** to **1000full nonegotiate**:

```
pixfirewall(config)# interface gb0 1000full nonegotiate
```

Sample output from the subsequent **show interface** command is as follows:

```
pixfirewall(config)# show interface gb0
interface gb-ethernet0 "intf2" is up, line protocol is down
Hardware is i82543 rev02 gigabit ethernet, address is 0003.47df.1e1c
MTU 1500 bytes, BW 1 Gbit full duplex, Force link-up
        5133 packets input, 628176 bytes, 0 no buffer
        Received 4202 broadcasts, 2 runts, 8 giants
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
        1832 packets output, 124948 bytes, 0 underruns
        input queue (curr/max blocks): hardware (41/128) software (0/2)
        output queue (curr/max blocks): hardware (0/2) software (0/4)
```

The "`Force link-up`" keyword indicates that the link was forced and not negotiated.

The following is sample output from the **show interface** command on a PIX 501. Notice that the interface speed and settings are always displayed as `100000 Kbit half duplex`.

```
pixfirewall(config)# show interface
interface ethernet0 "outside" is up, line protocol is up
  Hardware is i82559 ethernet, address is 0007.eb9b.56aa
  MTU 1500 bytes, BW 100000 Kbit half duplex
        114 packets input, 6840 bytes, 0 no buffer
        Received 114 broadcasts, 0 runts, 0 giants
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
        62982 packets output, 78915110 bytes, 0 underruns
        0 output errors, 0 collisions, 0 interface resets
        0 babbles, 0 late collisions, 0 deferred
        1483 lost carrier, 0 no carrier
        input queue (curr/max blocks): hardware (128/128) software (0/1)
        output queue (curr/max blocks): hardware (0/115) software (0/64)
interface ethernet1 "inside" is up, line protocol is up
  Hardware is i82559 ethernet, address is 0007.eb9b.56ab
  IP address 192.168.1.1, subnet mask 255.255.255.0
  MTU 1500 bytes, BW 100000 Kbit full duplex
        55005197 packets input, 903916376 bytes, 0 no buffer
        Received 0 broadcasts, 0 runts, 0 giants
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
        2 packets output, 120 bytes, 0 underruns
        0 output errors, 0 collisions, 0 interface resets
        0 babbles, 0 late collisions, 0 deferred
        0 lost carrier, 0 no carrier
        input queue (curr/max blocks): hardware (128/128) software (0/59)
        output queue (curr/max blocks): hardware (0/1) software (0/1)
```

Related Commands

nameif	Assigns a name to an interface.
ip address	Configures the IP address and mask for an interface, or defines a local address pool.

ip address

Identifies addresses for network interfaces, and enables you to set the number of times the PIX Firewall will poll for DHCP information.

[**no**] **ip address** *if_name ip_address* [*netmask*]

[**no**] **ip address outside dhcp** [**setroute**] [**retry** *retry_cnt*]

[**no**] **ip address** *if_name* **pppoe** [**setroute**]

[**no**] **ip address** *if_name ip_address netmask* **pppoe** [**setroute**]

clear ip

show ip

show ip address *if_name* **dhcp**

show ip address *if_name* **pppoe**

Syntax Description

clear ip	Clears all interface IP addresses. The **clear ip** command does not affect the **ip local pool** or **ip verify reverse-route** commands.
dhcp	Specifies PIX Firewall will use DHCP to poll for information. Enables the DHCP client feature on the specified interface.
if_name	The internal or external interface name designated by the **nameif** command.
ip_address	PIX Firewall unit's network interface IP address. Each interface IP address must be unique. Two or more interfaces must not be given the same IP address or IP addresses which are on the same IP network.
netmask	Network mask of *ip_address*.
outside	Interface from which the PIX Firewall will poll for information.
pppoe	Specifies to use Point-to-Point Protocol over Ethernet (PPPoE) to assign an IP address. The netmask of a dynamically retrieved address from PPPoE is 255.255.255.255.
retry	Enables PIX Firewall to retry a poll for DHCP information.
retry_cnt	Specifies the number of times PIX Firewall will poll for DHCP information. The values available are 4 to 16. If no value is specified, the default is 4.
setroute	This option tells the PIX Firewall to set the default route using the default gateway parameter the DHCP or PPPoE server returns.

Command Modes

Configuration mode.

Defaults

By default, the PIX Firewall will not retry to poll for DHCP information. The default value for *retry_cnt* is **4**.

Usage Guidelines

The **ip address** command lets you assign an IP address to each interface.

Note Each interface IP address must be unique and not on the same network as any another interface on the firewall.

Use the **show ip** command to view which addresses are assigned to the network interfaces. If you make a mistake while entering this command, reenter the command with the correct information. The **clear ip** command clears all interface IP addresses. The **clear ip** command does not affect the **ip local pool** or **ip verify reverse-route** commands.

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

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

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

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

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

Note If an IP address has not been configured for a physical or VLAN interface, or the IP address for the interface has been deleted using the **clear ip** command, the IP address for that interface is no longer set to 127.0.0.1 by default. In this case, the interface is does not have an IP address.

Note When using the IP address of an interface as the device ID in logging messages sent to a syslog server and the IP address of that interface is cleared, the device ID uses 0.0.0.0.

show ip address commands

The **show ip** command isplays IP addresses assigned to the network interfaces.

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

The **show ip address** *if_name* **pppoe** command displays detailed information about the PPPOE connection.

DHCP client

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

route as being set by a DHCP server. To reset the interface and delete the DHCP lease from PIX Firewall, configure a static IP address with the **ip address** *if_name ip_address* [*netmask*] or **ip address** *if_name* **pppoe | dhcp** [**setroute**] command, or use the **clear ip** command.

The **ip address dhcp** and **pppoe** command options are mutually exclusive.

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

PPPoE client

The PPPoE client functionality is turned off by default, and you must first use the **vpdn** commands to configure the PIX Firewall for PPPoE; the **vpdn** commands set the username, password, and authentication protocol for PPPoE access.

PPPoE is only supported on the PIX Firewall outside interface in PIX Firewall software Version 6.2.

The **ip address pppoe** command enables the PPPoE client feature within the PIX Firewall. (You can also use this command to clear and restart a PPPoE session; the current session shuts down and a new one restarts after entering this command.) You must enter the PPPoE configuration using the **vpdn** commands before enabling PPPoE with the **ip address pppoe** command.

You can also enable PPPoE by manually entering the IP address, using the **ip address** *if_name ip_address netmask* **pppoe** command. This command sets the PIX Firewall to use the specified address instead of negotiating with the PPPoE server to assign an address.

The **ip address setroute** command enables an access concentrator to set the default routes for the PPPoE client.

The **ip address pppoe** and **dhcp** command options are mutually exclusive.

For more information

See the *Cisco PIX Firewall and VPN Configuration Guide* for more information about the DHCP and PPPoE client features.

Examples

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

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

The Current IP Addresses are the same as the System IP Addresses on the failover active unit. When the primary unit fails, the Current IP Addresses become those of the standby unit.

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

```
show ip address outside dhcp
Temp IP Addr:209.165.201.57 for peer on interface:outside
Temp sub net mask:255.255.255.224
DHCP Lease server:209.165.200.225, state:3 Bound
DHCP Transaction id:0x4123
Lease:259200 secs, Renewal:129600 secs, Rebind:226800 secs
Temp default-gateway addr:209.165.201.1
```

```
Next timer fires after:111797 secs
Retry count:0, Client-ID:cisco-0000.0000.0000-outside

ip address outside dhcp retry 10
```

Related Commands

dhcpd	Configures the DHCP server.
vpdn	Configures VPDN (PPTP, L2TP, PPPoE) policy.

ip audit

Configures IDS signature use.

[**no**] **ip audit attack** [**action** [**alarm**] [**drop**] [**reset**]]

[**no**] **ip audit info** [action [alarm] [drop] [reset]]

[**no**] **ip audit interface** *if_name audit_name*

[**no**] **ip audit name** *audit_name* **attack** [**action** [**alarm**] [**drop**] [**reset**]]

[**no**] **ip audit name** *audit_name* **info** [**action** [**alarm**] [**drop**] [**reset**]]

[**no**] **ip audit signature** *signature_number* **disable**

show ip audit count [**global**] [**interface** *interface*]

show ip audit {**info | attack**}

show ip audit interface [*if_name*]

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

show ip audit signature [*signature_number*]

clear ip audit [**configuration**]

clear ip audit count [**global | interface** *interface*]

Syntax Description

action [**alarm**] [**drop**] [**reset**]	The **alarm** option reports to all configured syslog servers that a signature match is detected in a packet. 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**. When no option is specified (you enter "`ip audit info action`" only), all actions are disabled.
audit attack	Specify the default actions to be taken for attack signatures.
audit info	Specify the default actions to be taken for informational signatures or disable all actions.
audit interface	Apply an audit specification or policy (via the **ip audit name** command) to an interface.

audit name	Specify informational signatures, except those disabled or excluded by the **ip audit signature** command, as part of the policy.
audit signature	Specify which messages to display, attach a global policy to a signature, and disable or exclude a signature from auditing.
audit_name	Audit policy name viewed with the **show ip audit name** command.
clear	Resets **name, signature, interface, attack, info** to their default values**.**
configuration	The already configured **ip audit** commands.
count	The number of signature matches.
global	All firewall interfaces.
interface *interface*	The name of a firewall interface, defined by the **nameif** command.
signature_number	An IDS signature number.

Command Modes

Configuration mode.

Usage Guidelines

Cisco Intrusion Detection System (Cisco IDS) provides the following for IP-based systems:

- Traffic auditing. Application-level signatures will only be audited as part of an active session.
- Applies the audit to an interface.
- Supports different audit policies. Traffic matching a signature triggers a range of configurable actions.
- Disables the signature audit.
- Enables IDS and still disables actions of a signature class (informational, attack).

Auditing is performed by looking at the IP packets as they arrive at an input interface, if a packet triggers a signature and the configured action does not drop the packet, then the same packet can trigger other signatures.

PIX Firewall supports both inbound and outbound auditing.

For a complete list of supported Cisco IDS signatures, their wording, and whether they are attack or informational messages, refer to *Cisco PIX Firewall System Log Messages*.

Refer to the *Cisco Intrusion Prevention System* end-user guides for detailed information on each signature.

The **ip audit** commands are described in the sections that follow.

ip audit attack

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

The **no ip audit attack** command resets the action to be taken for attack signatures to the default action.

ip audit info

The **ip audit info** [**action** [**alarm**] [**drop**] [**reset**]] command specifies the default action to be taken for signatures classified as informational signatures. The **ip audit info action** command disables all actions. For example,

```
pixfirewall(config)# ip audit info action
Warning: no actions specified. All actions disabled.
```

The **no ip audit info** command sets the action to be taken for signatures classified as informational and reconnaissance to the default action.

ip audit interface

The **ip audit interface** *if_name audit_name* command applies an audit specification or policy (via the ip audit name command) to an interface. The **no ip audit interface** [*if_name*] command removes a policy from an interface.

ip audit name

The **ip audit name** *audit_name* **info** [**action** [**alarm**] [**drop**] [**reset**]] command specifies the informational signatures except those disabled or excluded by the **ip audit signature** command that are considered part of the policy. The **no ip audit name** *audit_name* [**info**] command removes the audit policy *audit_name*.

ip audit signature

The **ip audit signature** *signature_number* **disable** command specifies which messages to display, attaches a global policy to a signature, and disables or excludes a signature from auditing. The **no ip audit signature** *signature_number* command removes the policy from a signature. It is used to reenable a signature.

show ip audit commands

The **show ip audit attack** command displays the default attack actions.

The **show ip audit info** command displays the default informational actions.

The **show ip audit interface** command displays the interface configuration.

The **show ip audit name** command displays all audit policies or specific policies referenced by name and type where possible.

The **show ip audit signature** command displays disabled signatures.

Supported IDS Signatures

PIX Firewall lists the following single-packet IDS signature messages: 1000-1006, 1100, 1102, 1103, 2000-2012, 2150, 2151, 2154, 3040-3042, 4050-4052, 6050-6053, 6100-6103, 6150-6155, 6175, 6180, and 6190. All signature messages are not supported by PIX Firewall in this release. IDS syslog messages all start wit

PIX-4-4000*nn* and have the following format:

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

where the options are as follows:

sig_num	The signature number.
sig_msg	The signature message—approximately the same as the Cisco IDS signature message.

faddr	The IP address of the foreign host initiating the attack. (“Foreign” is relative; attacks can be perpetrated either from outside to an inside host, or from the inside to an outside host.)
laddr	The IP address of the local host to which the attack is directed. (“Local” is relative; attacks can be perpetrated either from the outside to an inside host, or from the inside to an outside host.)
int_name	The name of the interface on which the signature originated.

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
```

Examples

The following example disables the signature 6102 globally:

```
ip audit signature 6102 disable
```

The following example specifies default informational actions:

```
ip audit name attack1 info
```

The following example specifies an attack policy:

```
ip audit name attack2 attack action alarm drop reset
```

The following example applies a policy to an interface:

```
ip audit interface outside attack1
ip audit interface inside attack2
```

ip local pool

Identify addresses for a local pool.

ip local pool *pool_name pool_start_address[-pool_end_address] [mask mask]*

clear ip local pool *pool_name ip_address*[**-***ip_address*]

show ip local pool *pool_name ip_address*[**-***ip_address*]

[**no**] **ip local pool** *pool_name pool_start-address*[**-***pool_end-address*]

clear ip local pool	Removes all **ip local pool** configurations.
ip local pool	Creates a pool of local addresses to be used for assigning dynamic IP addresses to remote VPN clients. The address range of this pool of local addresses must not overlap with any command statement that lets you specify an IP address.
ip_address	Specify as a single IP address or use with *-ip_address* to specify a range of IP addresses.
-*ip_address*	Optional ending IP address.
no ip local pool	Deletes a local address pool.
pool_name	Local pool name.
pool_start_address *pool_end_address*	Local pool IP address range.
[mask <mask>]	Add an optional netmask. If the netmask is configured then the PIX Firewall headend will return it to the VPN client. If the netmask is not configured, PIX Firewall will retain backward compatibility with its previous behavior by not returning the netmask. If **netmask** is not configured, the PIX Firewall will use netmask 255.255.255.0.

Command Modes Configuration mode.

Usage Guidelines The **ip local pool** command lets you create a pool of local addresses to be used for assigning dynamic IP addresses to remote VPN clients. The address range of this pool of local addresses must not overlap with any command statement that lets you specify an IP address. To delete an address pool, use the **no ip local pool** command.

When a pool of addresses set by the **ip local pool** command is empty, the following syslog message appears:

```
%PIX-4-404101: ISAKMP: Failed to allocate address for client from pool poolname
```

To reference this pool of local addresses, use the **isakmp client configuration address-pool** command. Refer to the *Cisco PIX Firewall and VPN Configuration Guide* for information on the **isakmp** command.

```
pool_name ip_address [mask <mask>]
```

Because newer versions of VPN Client software might not work as expected without assigning a netmask directly through the PIX Firewall, this feature allows the user to assign the netmask to the VPN client. The operating system software will attempt to calculate what the netmask is if a netmask is not assigned, based on the class of the IP from "**ip local pool**". Using this default value may not be appropriate for network routing to work properly.

For example, 10.x.x.x. is a class A, and the windows routing table sets the netmask as 255.0.0.0.

The following example creates a pool of IP addresses and then displays the pool contents:

```
ip local pool mypool 10.0.0.10-10.0.0.20
show ip local pool mypool


Pool      Begin       End         Mask            Free        In use
mypool    10.0.0.10   10.0.0.20   Not configured  11          0

Available Addresses:
10.0.0.10
10.0.0.11
10.0.0.12
10.0.0.13
10.0.0.14
10.0.0.15
10.0.0.16
10.0.0.17
10.0.0.18
10.0.0.19
10.0.0.20
```

for **ip local pool** <*name*> <*range*> *[mask <mask>]*, a typical example range would be:

```
ip local pool <name> 10.1.1.1-10.1.1.254
```

ip verify reverse-path

Implements Unicast RPF IP spoofing protection.

ip verify reverse-path interface *int_name*

no ip verify reverse-path interface *int_name*

clear ip verify reverse-path interface *int_name*

clear ip verify

show ip verify [**reverse-path** [**interface** *int_name*]]

show ip verify statistics

Syntax Description

clear ip verify	Removes **ip verify** commands from the configuration.
clear ip verify reverse-path interface	Removes **ip verify reverse-path** commands for an individual interface from the configuration.
int_name	Name of an interface you want to protect from a DoS attack.
ip verify reverse-path interface	Protects an individual interface against IP spoofing by enabling both ingress and egress filtering to verify addressing and route integrity. See RFC 2267 for more information.
no ip verify reverse-path interface	Disables **ip verify reverse-path** filtering for an individual interface from the configuration.

Command Modes

Configuration mode.

Usage Guidelines

The **ip verify reverse-path** command is a security feature that does a route lookup based on the source address. Usually, the route lookup is based on the destination address. This is why it is called reverse path forwarding. With this command enabled, packets are dropped if there is no route found for the packet or the route found does not match the interface on which the packet arrived.

The **ip verify reverse-path** command lets you specify which interfaces to protect from an IP spoofing attack using network ingress and egress filtering, which is described in RFC 2267. This command is disabled by default and provides Unicast Reverse Path Forwarding (Unicast RPF) functionality for the PIX Firewall.

The **clear ip verify** command removes **ip verify** commands from the configuration. Unicast RPF is a unidirectional input function that screens inbound packets arriving on an interface. Outbound packets are not screened.

Because of the danger of IP spoofing in the IP protocol, measures need to be taken to reduce this risk when possible. Unicast RPF, or reverse route lookup, prevents such manipulation under certain circumstances.

The **ip verify reverse-path** command provides both ingress and egress filtering. Ingress filtering checks inbound packets for IP source address integrity, and is limited to addresses for networks in the enforcing entity's local routing table. If the incoming packet does not have a source address represented by a route, then it is impossible to know whether the packet has arrived on the best possible path back to its origin. This is often the case when routing entities cannot maintain routes for every network.

Egress filtering verifies that packets destined for hosts outside the managed domain have IP source addresses verifiable by routes in the enforcing entity's local routing table. If an exiting packet does not arrive on the best return path back to the originator, then the packet is dropped and the activity is logged.

Egress filtering prevents internal users from launching attacks using IP source addresses outside of the local domain because most attacks use IP spoofing to hide the identity of the attacking host. Egress filtering makes the task of tracing the origin of an attack much easier. When employed, egress filtering enforces what IP source addresses are obtained from a valid pool of network addresses. Addresses are kept local to the enforcing entity and are therefore easily traceable.

Unicast RPF is implemented as follows:

- ICMP packets have no session, so each packet is checked.
- UDP and TCP have sessions, so the initial packet requires a reverse route lookup. Subsequent packets arriving during the session are checked using an existing state maintained as part of the session. Non-initial packets are checked to ensure they arrived on the same interface used by the initial packet.

Note Before using this command, add static **route** command statements for every network that can be accessed on the interfaces you wish to protect. Only enable this command if routing is fully specified. Otherwise, PIX Firewall will stop traffic on the interface you specify if routing is not in place.

Use the **show interface** command to view the number dropped packets, which appears in the "unicast rpf drops" counter.

Examples

The following example protects traffic between the inside and outside interfaces and provides **route** command statements for two networks, 10.1.2.0 and 10.1.3.0, that connect to the inside interface via a hub:

```
ip address inside 10.1.1.1 255.255.0.0
route inside 10.1.2.0 255.255.0.0 10.1.1.1 1
route inside 10.1.3.0 255.255.0.0 10.1.1.1 1
ip verify reverse-path interface outside
ip verify reverse-path interface inside
```

The **ip verify reverse-path interface outside** command statement protects the outside interface from network ingress attacks from the Internet, whereas the **ip verify reverse-path interface inside** command statement protects the inside interface from network egress attacks from users on the internal network.

The following is sample output from the **show ip verify statistics** and **clear ip verify statistics** commands:

```
pixfirewall(config)# show ip verify statistics
interface outside: 2 unicast rpf drops
interface inside: 1 unicast rpf drops
interface intf2: 3 unicast rpf drops

pixfirewall(config)# clear ip verify statistics

pixfirewall(config)# show ip verify statistics
interface outside: 0 unicast rpf drops
interface inside: 0 unicast rpf drops
interface intf2: 0 unicast rpf drops
```

isakmp

Configures the Internet Security Association Key Management Protocol (ISAKMP) for IPSec Internet Key Exchange (IKE). See also the **isakmp policy** command.

[**no**] **isakmp client configuration address-pool local** *pool-name* [*interface-name*]

[**no**] **isakmp enable** *interface-name*

[**no**] **isakmp identity** {**address | hostname |** [**key-id** *key_id_string*]}

isakmp keepalive *seconds* [*retry_seconds*]

[**no**] **isakmp key** *keystring* **address** *peer-address* [**netmask** *mask*] [**no-xauth**] [**no-config-mode**]

isakmp log <#events>

isakmp nat-traversal [*natkeepalive*]

[**no**] **isakmp peer fqdn** *fqdn* **no-xauth no-config-mode**

clear [**crypto**] **isakmp sa**

clear isakmp

show isakmp identity

show isakmp sa [**detail**]

Syntax Description

address	The IP address of the host exchanging ISAKMP identity information.
fqdn *fqdn*	The fully qualified domain name of the peer. This is used to identify a peer that is a security gateway.
hostname	The name of the host exchanging ISAKMP identity information.
interface-name	The name of the interface on which to enable ISAKMP negotiation.
keepalive *seconds*	The keepalive interval can be between 10 and 3600 seconds. The retry interval can be between 2 and 10 seconds, with the default being 2 seconds. The retry interval is the interval between retries after a keepalive response has not been received. You can specify the keepalive interval without specifying the retry interval, but cannot specify the retry interval without specifying the keepalive interval.
key	Specifies the authentication pre-shared key. Use any combination of alphanumeric characters up to 128 bytes. This pre-shared key must be identical at both peers.
key-id *key_id_string*	String used by the remote peer to look up the pre-shared key. (This is intended for use with third-party VPN headend devices that do not support the Unity protocol.)
log	Sets the size of the log buffer.
<#events>	(Min 0, max 50,000, default 0 (disabled); each event uses 20 bytes)
netmask *mask*	(Optional) The netmask of 0.0.0.0. can be entered as a wildcard indicating the key could be used for any peer that does not have a key associated with its specific IP address.
natkeepalive	Sets the NAT keep alive interval, from 10 to 3600 seconds. The default is 20 seconds.
nat-traversal	Turns on or off NAT traversal. (NAT traversal is off by default.)
no-config-mode	This is only to be used if you enabled the IKE Mode Configuration feature, and you have an IPSec peer that is a gateway. This option associates a given pre-shared key with a gateway and allows an exception to the IKE Mode Configuration feature enabled by the **crypto map client configuration address** command.

no-xauth	This is only to be used if you enabled the Xauth feature, and you have an IPSec peer that is a gateway. This option associates a given pre-shared key with a gateway and allows an exception to the Xauth feature enabled by the **crypto map client authentication** command.
peer-address	Specifies the IPSec peer's IP address for the pre-shared key.
pool-name	Specify the name of a local address pool to allocate the dynamic client IP.
retry_seconds	Specifies the time interval before a keepalive message is sent if a keepalive response is not received from the previous request.

Command Modes

Configuration mode.

Defaults

By default, NAT traversal (**isakmp nat-traversal**) is disabled.

The default ISAKMP identity is **isakmp identity hostname**.

Usage Guidelines

The **show isakmp identity** command displays the current ISAKMP identity.

The **show isakmp sa** command displays all current IKE security associations between the PIX Firewall and its peer.

The sections that follow describe each **isakmp** command.

isakmp client configuration address-pool local

The **isakmp client configuration address-pool local** command is used to configure the IP address local pool to reference IKE. Use the **no crypto isakmp client configuration address-pool local** command to restore to the default value.

Before using this command, use the **ip local pool** command to define a pool of local addresses to be assigned to a remote IPSec peer.

Examples

The following example references IP address local pools to IKE with "mypool" as the pool-name:

```
isakmp client configuration address-pool local mypool outside
```

isakmp enable

Use the **isakmp enable** *interface-name* command to enable ISAKMP negotiation on the interface on which the IPSec peer will communicate with the PIX Firewall. Use the **no isakmp enable** command to disable IKE.

The following example shows how to disable IKE on the inside interface:

```
no isakmp enable inside
```

isakmp identity

To define the ISAKMP identity the PIX Firewall uses when participating in the IKE protocol, use the **isakmp identity** command. Use **no isakmp identity** command to reset the ISAKMP identity to the default value. The default ISAKMP identity is **hostname**.

When two peers use IKE to establish IPSec security associations, each peer sends its ISAKMP identity to the remote peer. It will send either its IP address or host name depending on how each has its ISAKMP identity set. By default, the PIX Firewall unit's ISAKMP identity is set to the IP address. As a general rule, set the PIX Firewall and its peer's identities in the same way to avoid an IKE negotiation failure. This failure could be due to either the PIX Firewall or its peer not recognizing its peer's identity.

Note If you are using RSA signatures as your authentication method in your IKE policies, we recommend that you set each participating peer's identity to **hostname**. Otherwise, the ISAKMP security association to be established during Phase 1 of IKE may fail.

The following example uses pre-shared keys between the two PIX Firewall units (PIX Firewall 1 and PIX Firewall 2) that are peers, and sets both their ISAKMP identities to host name.

At the PIX Firewall 1, the ISAKMP identity is set to **hostname**:

```
isakmp identity hostname
```

At the PIX Firewall 2, the ISAKMP identity is set to **hostname**:

```
isakmp identity hostname
```

isakmp identity key-id

The **isakmp identity key-id** *key_id_string* command sends the specified *key_id_string* using aggressive mode. This is intended to enable third-party VPN headend devices that do not support the Unity protocol to interoperate with a DHCP-enabled firewall at a remote site.

Note If the VPN client feature is enabled on the firewall, the **vpnclient** group name takes precedence over the **isakmp identity key-id** setting, and the firewall sends **vpnclient** group name as the **key-id**.

isakmp keepalive

The **isakmp keepalive** *seconds* [*retry_seconds*] command sets the keepalive lifetime interval. The keepalive interval can be between 10 and 3600 seconds. The retry interval can be between 2 and 10 seconds, with the default being 2 seconds. The retry interval is the interval between retries after a keepalive response has not been received. You can specify the keepalive lifetime interval without specifying the retry interval, but cannot specify the retry interval without specifying the keepalive lifetime interval.

isakmp key address

To configure a pre-shared authentication key and associate the key with an IPSec peer address or host name, use the **isakmp key address** command. Use the **no isakmp key address** command to delete a pre-shared authentication key and its associated IPSec peer address.

You would configure the pre-shared key at both peers whenever you specify pre-shared key in an IKE policy. Otherwise, the policy cannot be used because it will not be submitted for matching by the IKE process.

A netmask of 0.0.0.0. can be entered as a wildcard indicating that any IPSec peer with a given valid pre-shared key is a valid peer.

Note The PIX Firewall or any IPSec peer can use the same authentication key with multiple peers, but this is not as secure as using a unique authentication key between each pair of peers.

Configure a pre-shared key associated with a given security gateway to be distinct from a wildcard, pre-shared key (pre-shared key plus a netmask of 0.0.0.0) used to identify and authenticate the remote VPN clients.

The no-xauth or no-config-mode command options are to be used only if the following criteria are met:

- You are using the pre-shared key authentication method within your IKE policy.
- The security gateway and VPN client peers terminate on the same interface.
- The Xauth or IKE Mode Configuration feature is enabled for VPN client peers.

The **isakmp key** *keystring* **address** *ip-address* [**no-xauth**] [**no-config-mode**] command lets you configure a pre-shared authentication key, associate the key with a given security gateway's address, and make an exception to the enabled Xauth feature, IKE Mode Configuration feature, or both (the most common case) for this peer.

Both the Xauth and IKE Mode Configuration features are specifically designed for remote VPN clients. The Xauth feature allows the PIX Firewall to challenge the peer for a username and password during IKE negotiation. The IKE Mode Configuration enables the PIX Firewall to download an IP address to the peer for dynamic IP address assignment. Most security gateways do not support the Xauth and IKE Mode Configuration features.

You cannot enable Xauth or IKE Mode Configuration on a interface when terminating an L2TP/IPSec tunnel using the Microsoft L2TP/IPSec client v1.0 (which is available on Windows NT, Windows XP, Windows 98 and Windows ME OS). Instead, you can do either of the following:

- Use a Windows 2000 L2TP/IPSec client, or
- Use the **isakmp key** *keystring* **address** *ip-address* **netmask** *mask* **no-xauth no-config-mode** command to exempt the L2TP client from Xauth and IKE Mode Configuration. However, if you exempt the L2TP client from Xauth or IKE Mode Configuration, all the L2TP clients must be grouped with the same ISAKMP pre-shared key or certificate and have the same fully qualified domain name.

If you have the **no-xauth** command option configured, the PIX Firewall will not challenge the peer for a username and password. Similarly, if you have the **no-config-mode** command option configured, the PIX Firewall will not attempt to download an IP address to the peer for dynamic IP address assignment.

Use the **no key** *keystring* **address** *ip-address* [**no-xauth**] [**no-config-mode**] command to disable the **key** *keystring* **address** *ip-address* [**no-xauth**] [**no-config-mode**] command that you previously enabled.

See the **crypto map client authentication** command within the **crypto map** command page for more information about the Xauth feature. See the **crypto map client configuration address** command within the **crypto map** command page for more information about the IKE Mode Config feature.

The following example shows "sharedkeystring" as the authentication key to share between the PIX Firewall and its peer specified by an IP address of 10.1.0.0:

```
isakmp key sharedkeystring address 10.1.0.0
```

The following example shows use of a wildcard, pre-shared key. The "sharedkeystring" is the authentication key to share between the PIX Firewall and its peer (in this case a VPN client) specified by an IP address of 0.0.0.0. and a netmask of 0.0.0.0.

```
isakmp key sharedkeystring address 0.0.0.0 netmask 0.0.0.0
```

The following example shows use of the command options **no-xauth and no-config-mode** in relation to three PIX Firewall peers that are security gateways. These security gateways terminate IPSec on the same interface as the VPN clients. Both the Xauth and IKE Mode Config features are enabled. This

means there is a need to make an exception to these two features for each security gateway. The example shows each security gateway peer has a unique pre-shared key to share with the PIX Firewall. The peers' IP addresses are 10.1.1.1, 10.1.1.2, 10.1.1.3, and the netmask of 255.255.255.255 is specified.

```
isakmp key secretkey1234 address 10.1.1.1 netmask 255.255.255.255 no-xauth
    no-config-mode
isakmp key secretkey4567 address 10.1.1.2 netmask 255.255.255.255 no-xauth
    no-config-mode
isakmp key secretkey7890 address 10.1.1.3 netmask 255.255.255.255 no-xauth
    no-config-mode
```

isakmp log <#events>

A circular event tracing buffer has been added to assist in troubleshooting when syslogs are unavailable. The event buffer is disabled by default. After it is enabled, the following events will be recorded in the buffer.

```
    DPD_TX
    DPD_RX
    DPD_ACK_TX
    DPD_ACK_RX
    DPD_FAIL
    P1_RETRAN
    P2_RETRAN
    VPNC_CONNECT
    VPNC_DISCONNECT
    IKE_DELETE_TX
    IKE_DELETE_RX
    MALFORMED_PAYLOAD
    DUPLICATE_PACKET
    P1_INIT
    P1_RESP
    P1_DONE
    P2_INIT
    P2_RESP
    P2_DONE
    P1_REKEY
```

```
mypix# show isakmp log


    16:18:31.964 UTC Fri May 21 2004, peer 63.67.72.161, P2_INIT
    16:18:31.774 UTC Fri May 21 2004, peer 63.67.72.161, VPNC_CONNECT
    16:18:31.774 UTC Fri May 21 2004, peer 63.67.72.161, P1_DONE
    16:18:30.234 UTC Fri May 21 2004, peer 63.67.72.161, P1_INIT
    02:11:23.762 UTC Fri May 21 2004, peer 63.67.72.161, IKE_DELETE_TX
    02:11:22.982 UTC Fri May 21 2004, peer 63.67.72.161, P1_RESP
    02:11:22.762 UTC Fri May 21 2004, peer 63.67.72.161, VPNC_DISCONNECT
    02:11:22.692 UTC Fri May 21 2004, peer 63.67.72.161, P2_RETRAN (1)
    02:11:22.682 UTC Fri May 21 2004, peer 63.67.72.161, P2_RETRAN (1)
    02:11:22.592 UTC Fri May 21 2004, peer 63.67.72.161, P2_RETRAN (1)
    02:11:21.702 UTC Fri May 21 2004, peer 63.67.72.161, P2_DONE
    02:11:21.682 UTC Fri May 21 2004, peer 63.67.72.161, P2_DONE
    02:11:21.432 UTC Fri May 21 2004, peer 63.67.72.161, P2_RESP
    02:11:21.422 UTC Fri May 21 2004, peer 63.67.72.161, P2_RESP
    02:11:18.782 UTC Fri May 21 2004, peer 63.67.72.161, DPD_TX (1074122247)
    02:11:16.701 UTC Fri May 21 2004, peer 63.67.72.161, P2_DONE
    02:11:16.321 UTC Fri May 21 2004, peer 63.67.72.161, P2_RESP
    02:11:13.781 UTC Fri May 21 2004, peer 63.67.72.161, DPD_TX (1074122246)
    02:11:13.141 UTC Fri May 21 2004, peer 63.67.72.161, P2_RESP
```

isakmp nat-traversal

Network Address Translation (NAT), including Port Address Translation (PAT), is used in many networks where IPSec is also used, but there are a number of incompatibilities that prevent IPSec packets from successfully traversing NAT devices. NAT traversal enables ESP packets to pass through one or more NAT devices.

The firewall supports NAT traversal as described by Version 2 and Version 3 of the IETF "UDP Encapsulation of IPsec Packets" draft, available at http://www.ietf.org/html.charters/ipsec-charter.html, and NAT traversal is supported for both dynamic and static crypto maps. NAT traversal is disabled by default on the firewall.

To enable NAT traversal, check that ISAKMP is enabled (you can enable it with the **isakmp enable** *if_name* command) and then use the **isakmp nat-traversal** [*natkeepalive*] command. (This command appears in the configuration if both ISAKMP is enabled and NAT traversal is enabled.) If you have enabled NAT traversal, you can disable it with the **no isakmp nat-traversal** command. Valid values for *natkeepalive* are from 10 to 3600 seconds. The default is 20 seconds.

If needed, the **show isakmp sa detail** command assists in debugging NAT traversal.

isakmp peer fqdn no-xauth | no-config-mode

The **isakmp peer fqdn** *fqdn* **no-xauth | no-config-mode** command is to be used only if the following criteria are met:

- You are using the RSA signatures authentication method within your IKE policy.
- The security gateway and VPN client peers terminate on the same interface.
- The Xauth or IKE Mode Configuration feature is enabled for VPN client peers.

The **isakmp peer fqdn** *fqdn* **no-xauth | no-config-mode** command lets you identify a peer that is a security gateway and make an exception to the enabled Xauth feature, IKE Mode Configuration feature, or both (the most common case) for this peer.

Both the Xauth and IKE Mode Configuration features are specifically designed for remote VPN clients. The Xauth feature allows the PIX Firewall to challenge the peer for a username and password during IKE negotiation. The IKE Mode Configuration feature enables the PIX Firewall to download an IP address to the peer for dynamic IP address assignment. Most security gateways do not support the Xauth and IKE Mode Configuration features.

If you have the **no-xauth** command option configured, the PIX Firewall will not challenge the peer for a username and password. Similarly, if you have the **no-config-mode** command option configured, the PIX Firewall will not attempt to download an IP address to the peer for dynamic IP address assignment.

Note If you are using RSA signatures as your authentication method in your IKE policies, we recommend that you set each participating peer's identity to hostname using the **isakmp identity hostname** command. Otherwise, the ISAKMP security association to be established during Phase 1 of IKE may fail.

Use the **no isakmp peer fqdn** *fqdn* **no-xauth | no-config-mode** command to disable the **isakmp peer fqdn** *fqdn* **no-xauth | no-config-mode** command that you previously enabled.

See the **crypto map client authentication** within the **crypto map** command page for more information about the Xauth feature. See the **crypto map client configuration address** command within the **crypto map** command page for more information about the IKE Mode Config feature.

The following example shows use of the command options **no-xauth and no-config-mode** in relation to three PIX Firewall peers that are security gateways. These security gateways terminate IPSec on the same interface as the VPN clients. Both the Xauth and IKE Mode Config features are enabled. This means there is a need to make an exception to these two features for each security gateway. Each security gateway peer's fully qualified domain name is specified.

```
isakmp peer fqdn hostname1.example.com no-xauth no-config-mode
isakmp peer fqdn hostname2.example.com no-xauth no-config-mode
isakmp peer fqdn hostname3.example.com no-xauth no-config-mode
```

show isakmp sa

To view all current IKE security associations between the PIX Firewall and its peer, use the **show isakmp sa** command.

The following is sample output from the **show isakmp sa** command after IKE negotiations were successfully completed between the PIX Firewall and its peer:

```
pixfirewall# show isakmp sa

dst             src             state      pending   created
16.132.40.2     16.132.30.2     QM_IDLE    0         1
```

The following is sample output from the **show isakmp sa detail** command (used for debugging NAT traversal):

```
pixfirewall# show isakmp sa detail
Total     : 1
Embryonic : 0

Local               Remote               Encr     Hash    Auth    State      Lifetime
192.168.10.2:4500   192.168.10.5:1178    3des     sha     psk     QM_IDLE    117
```

`Local` is the IP address and port of the firewall on which the command is run (the format is IP_Address:port); `Remote` is the peer IP address and port; `Encr` is the encryption algorithm; `Hash` is the hash algorithm; `Auth` is the authorization method, preshared key, or rsa; `State` is the state of the connection, and `Lifetime` is either the time until re-key or until expiration and deletion.

clear isakmp

The **clear isakmp** command removes all **isakmp** command statements from the configuration.

clear [crypto] isakmp sa

The **clear** [**crypto**] **isakmp sa** command deletes active IKE security associations. The keyword **crypto** is optional.

isakmp policy

Configures specific Internet Key Exchange (IKE) algorithms and parameters, within the IPSec Internet Security Association Key Management Protocol (ISAKMP) framework, for the Authentication Header (AH) and Encapsulating Security Payload (ESP) IPSec protocols. See also the **isakmp** command.

[**no**] **isakmp policy** *priority* **authentication** *pre-share* | *rsa-sig*

[**no**] **isakmp policy** *priority* **encryption aes | aes-192| aes-256 | des | 3des**

[**no**] **isakmp policy** *priority* **group 1 | 2 | 5**

[**no**] **isakmp policy** *priority* **hash** *md5* | *sha*

[**no**] **isakmp policy** *priority* **lifetime** *seconds*

show isakmp policy

Syntax Description

3des	Specifies that the Triple DES encryption algorithm is to be used in the IKE policy.
aes	Selecting this option means that encrypted IKE messages protected by this suite are encrypted using AES with a 128-bit key.
aes-192	Selecting this option means that encrypted IKE messages protected by this suite are encrypted using AES with a 192-bit key.
aes-256	Selecting this option means that encrypted IKE messages protected by this suite are encrypted using AES with a 256-bit key.
des	Specifies 56-bit DES-CBC as the encryption algorithm to be used in the IKE policy.
group 1	Specify that the 768-bit Diffie-Hellman group is to be used in the IKE policy. This is the default value.
group 2	Specifies that the 1024-bit Diffie-Hellman group 2 be used in the IKE policy.
group 5	Specifies that the 1536-bit Diffie-Hellman group 5 be used in the IKE policy.
lifetime *seconds*	Specify how many seconds each security association should exist before expiring. To propose a finite lifetime, use an integer from 120 to 86,400 seconds (one day). Specify 0 seconds for infinite lifetime.
md5	Specify MD5 (HMAC variant) as the hash algorithm to be used in the IKE policy.
pre-share	Specify pre-shared keys as the authentication method.
priority	Uniquely identifies the Internet Key Exchange (IKE) policy and assigns a priority to the policy. Use an integer from 1 to 65,534, with 1 being the highest priority and 65,534 the lowest.
rsa-sig	Specify RSA signatures as the authentication method. RSA signatures provide non-repudiation for the IKE negotiation. This basically means you can prove to a third party whether you had an IKE negotiation with the peer.
sha	Specify SHA-1 (HMAC variant) as the hash algorithm to be used in the IKE policy. This is the default hash algorithm.

Command Modes

Configuration mode.

Defaults

The default ISAKMP policy encryption is **des**.

The default hash algorithm is SHA-1 (HMAC variant).

Usage Guidelines

The **isakmp policy** command lets you negotiate IPSec security associations and enable IPSec secure communications.

The following is an example of the **isakmp policy** command:

```
isakmp policy 93 group 2
```

Note The Cisco VPN Client Version 3.x requires **isakmp policy** to have DH **group 2** configured. (If you have DH **group 1** configured, the Cisco VPN Client cannot connect.)

AES support is available on firewalls licensed for VPN-3DES only. Due to the large key sizes provided by AES, ISAKMP negotiation should use Diffie-Hellman (DH) **group 5** instead of **group 1** or **group 2**. This is done with the **isakmp policy priority group 5** command.

The **show isakmp policy** command displays parameters for each IKE policy, including defaults.

isakmp policy authentication

The isakmp policy authentication command lets you specify the authentication method within an IKE policy. IKE policies define a set of parameters to be used during IKE negotiation.

If you specify RSA signatures, you must configure the PIX Firewall and its peer to obtain certificates from a CA. If you specify pre-shared keys, you must separately configure these pre-shared keys within the PIX Firewall and its peer.

Use the **no isakmp policy authentication** command to reset the authentication method to the default value of RSA signatures.

The following example shows use of the **isakmp policy authentication** command. This example sets the authentication method of rsa-signatures to be used within the IKE policy with the priority number of 40.

```
isakmp policy 40 authentication rsa-sig
```

isakmp policy encryption

To specify the encryption algorithm to be used within an IKE policy, use the **isakmp policy encryption** command. AES with a 128-bit key (**aes**), AES with a 192-bit key (**aes-192**), AES with a 256-bit key (**aes-256**), DES (**des**), and 3DES (**3des**) are the supported encryption algorithms. (IKE policies define the set of parameters to be used during IKE negotiation.)

Use the **no isakmp policy encryption** command to reset the encryption algorithm to the default value, which is **des**.

The following example shows use of the **isakmp policy encryption** command; it sets 128-bit key AES encryption as the algorithm to be used within the IKE policy with the priority number of 25.

```
isakmp policy 25 encryption aes
```

The following example sets the 3DES algorithm to be used within the IKE policy with the priority number of 40.

```
isakmp policy 40 encryption 3des
```

isakmp policy group

Use the **isakmp policy group** command to specify the Diffie-Hellman group to be used in an IKE policy. IKE policies define a set of parameters to be used during IKE negotiation.

There are three group options: 768-bit (DH Group 1), 1024-bit (DH Group 2), or 1536-bit (DH Group 5). The 1024-bit and 1536-bit Diffie-Hellman Groups provide stronger security, but it require more CPU time to execute.

Use the **no isakmp policy group** command to reset the Diffie-Hellman group identifier to the default value of group 1 (768-bit Diffie Hellman).

The following example shows use of the **isakmp policy group** command. This example sets group 2, the 1024-bit Diffie Hellman, to be used within the IKE policy with the priority number of 40.

```
isakmp policy 40 group 2
```

Note Cisco VPN Client Version 3.x uses Diffie-Hellman group 2 and Cisco VPN Client 3000 Version 2.5/2.6 uses Diffie-Hellman group 1. If you are using Cisco VPN Client Version 3.x, configure Diffie-Hellman group 2 by using the **isakmp policy group 2** command.

isakmp policy hash

Use the **isakmp policy hash** command to specify the hash algorithm to be used in an IKE policy. IKE policies define a set of parameters to be used during IKE negotiation.

There are two hash algorithm options: SHA-1 and MD5. MD5 has a smaller digest and is considered to be slightly faster than SHA-1.

To reset the hash algorithm to the default value of SHA-1, use the **no isakmp policy hash** command.

The following example shows use of the **isakmp policy hash** command. This example sets the MD5 hash algorithm to be used within the IKE policy with the priority number of 40.

```
isakmp policy 40 hash md5
```

isakmp policy lifetime

To specify the lifetime of an IKE security association before it expires, use the **isakmp policy lifetime** command. An infinite lifetime can also be specified in case the peer does not propose a lifetime. Use the **no isakmp policy lifetime** command to reset the security association lifetime to the default value of 86,400 seconds (one day).

When IKE begins negotiations, it looks to agree upon the security parameters for its own session. The agreed-upon parameters are then referenced by a security association at each peer. The security association is retained by each peer until the security association's lifetime expires. Before a security association expires, it can be reused by subsequent IKE negotiations, which can save time when setting up new IPSec security associations. New security associations are negotiated before current security associations expire.

To save setup time for IPSec, configure a longer IKE security association lifetime. However, the shorter the lifetime (up to a point), the more secure the IKE negotiation is likely to be.

Note If the IKE security association is set to an infinite lifetime, but the peer proposes a finite lifetime, then the negotiated finite lifetime from the peer will be used.

Note When PIX Firewall initiates an IKE negotiation between itself and an IPSec peer, an IKE policy can be selected only if the lifetime of the peer's policy is shorter than or equal to the lifetime of its policy. Then, if the lifetimes are not equal, the shorter lifetime will be selected.

The following example shows use of the **isakmp policy lifetime** command. This example sets the lifetime of the IKE security association to 50,400 seconds (14 hours) within the IKE policy with the priority number of 40.

```
isakmp policy 40 lifetime 50400
```

The following example sets the IKE security association to an infinite lifetime.

```
isakmp policy 40 lifetime 0
```

show isakmp policy

To view the parameters for each IKE policy including the default parameters, use th**e show isakmp policy** command.

The following is sample output from the **show isakmp policy** command after two IKE policies were configured (with priorities 70 and 90 respectively):

```
show isakmp policy

Protection suite priority 70
     encryption algorithm:   DES - Data Encryption Standard (56 bit keys)
     hash algorithm: Message Digest 5
     authentication method:  Rivest-Shamir-Adleman Signature
     Diffie-Hellman group:   #2 (1024 bit)
     lifetime:     5000 seconds, no volume limit
Protection suite priority 90
     encryption algorithm:   DES - Data Encryption Standard (56 bit keys)
     hash algorithm: Secure Hash Standard
     authentication method:  Pre-Shared Key
     Diffie-Hellman group:   #1 (768 bit)
     lifetime:    10000 seconds, no volume limit
Default protection suite
     encryption algorithm:   DES - Data Encryption Standard (56 bit keys)
     hash algorithm: Secure Hash Standard
     authentication method:  Rivest-Shamir-Adleman Signature
     Diffie-Hellman group:   #1 (768 bit)
     lifetime:    86400 seconds, no volume limit
```

Note Although the output shows "no volume limit" for the lifetimes, you can currently only configure a time lifetime (such as 86,400 seconds) or infinity; volume limit lifetimes are not currently configurable.

Examples

The following is sample output from the **show isakmp** and **show isakmp policy** commands for a configuration using Diffie-Hellman group 5 in its ISAKMP policy:

```
pixfirewall(config)# show isakmp
isakmp enable outside
isakmp key ******** address 0.0.0.0 netmask 0.0.0.0
isakmp policy 1 authentication pre-share
isakmp policy 1 encryption 3des
isakmp policy 1 hash md5
isakmp policy 1 group 5
isakmp policy 1 lifetime 86400

pixfirewall(config)# show isakmp policy
Protection suite of priority 8
        encryption algorithm:   Three key triple DES
        hash algorithm:         Message Digest 5
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #5 (1536 bit)
        lifetime:               86400 seconds, no volume limit
```

Related Commands

ca	For Certificate Enrollment Protocol (CEP), creates and enrolls RSA key pairs into a Public Key Infrastructure (PKI).
crypto dynamic-map	Configures the IPSec crypto dynamic-map policy.
crypto ipsec	Configures the transform set and IPSec security association (SA) lifetime.
cyrpto map	Configures the IPsec crypto map policy.

kill

Terminate a Telnet session.

kill *telnet_id*

Syntax Description

telnet_id	Telnet session ID.

Command Modes

Privileged mode.

Usage Guidelines

The **kill** command terminates a Telnet session. Use the **who** command to view the Telnet session ID value. When you kill a Telnet session, the PIX Firewall lets any active commands terminate and then drops the connection without warning the user.

Examples

The following is sample output from the **show who** command, which is used to list the active Telnet sessions, and the use of the **kill** command to end Telnet session 2:

```
show who
2: From 10.10.54.0
kill 2
```

Related Commands

who	Shows the active administration sessions on the firewall.
telnet	Adds Telnet access to the firewall console and sets the idle timeout.

logging

Enable or disable syslog and SNMP logging.

[**no**] **logging on**

[**no**] **logging buffered** *level*

[**no**] **logging console** *level*

logging device-id {**hostname** | **ipaddress** *if_name* | **string** *text*}

no logging device-id

[**no**] **logging facility** *facility*

[**no**] **logging history** *level*

[**no**] **logging host** [*in_if_name*] *ip_address* [*protocol*/*port*] [**format emblem**]

[**no**] **logging message** *syslog_id* [**level** *level*]

[**no**] **logging monitor** *level*

[**no**] **logging queue** *queue_size*

[**no**] **logging standby**

[**no**] **logging timestamp**

[**no**] **logging trap** *level*

clear logging [**disable**]

show logging [**message** {*syslog_id* | **all**} | **level** | **disabled**]

show logging queue

Syntax Description

all	All syslog message IDs.
buffered	Send syslog messages to an internal buffer that can be viewed with the **show logging** command. Use the **clear logging** command to clear the message buffer. New messages append to the end of the buffer.
clear	Clear the buffer for use with the **logging buffered** command.

console	Specify that syslog messages appear on the PIX Firewall console as each message occurs. You can limit the types of messages that appear on the console with *level*. We recommend that you do not use this command in production mode because its use degrades PIX Firewall performance.
device-id	The device ID of the PIX Firewall to include in the syslog message.
disabled	Clear or display suppressed messages. You can suppress messages with the **no logging message** command.
facility	Specify the syslog facility. The default is 20.
facility	Eight facilities LOCAL0(16) through LOCAL7(23); the default is LOCAL4(20). Hosts file the messages based on the *facility* number in the message.
format emblem	This option enables EMBLEM format logging on a per-syslog-server basis. EMBLEM format logging is available for UDP syslog messages only and is disabled by default.
history	Set the SNMP message level for sending syslog traps.
host	Specify a syslog server that will receive the messages sent from the PIX Firewall. You can use multiple **logging host** commands to specify additional servers that would all receive the syslog messages. A PIX Firewall Syslogs Server can be configured to receive syslogs over UDP or TCP, not both. Likewise the PIX Firewall can send either UDP or TCP syslog messages to the PIX Firewall Syslog Server.
hostname	Specifies to use the host name of the PIX Firewall to uniquely identify the syslog messages from the PIX Firewall.
if_name	Specifies the name of the interface whose IP address is used to uniquely identify the syslog messages from the PIX Firewall.
in_if_name	Interface on which the syslog server resides.
ip_address	Syslog server's IP address.
ipaddress	Specifies to use the IP address of the specified PIX Firewall interface to uniquely identify the syslog messages from the PIX Firewall.
level	Specify the syslog message level as a number or string. The *level* you specify means that you want that *level* and those less than the *level*. For example, if *level* is **3**, syslog displays **0**, **1**, **2**, and **3** messages**.** Possible number and string *level* values are: • **0**—**emergencies**—System unusable messages • **1**—**alerts**—Take immediate action • **2**—**critical**—Critical condition • **3**—**errors**—Error message • **4**—**warnings**—Warning message • **5**—**notifications**—Normal but significant condition • **6**—**informational**—Information message • **7**—**debugging**—Debug messages and log FTP commands and WWW URLs
message	Specify a message to be allowed. Use the **no logging message** command to suppress a syslog message. Use the **clear logging disabled** command to reset the disallowed messages to the original set. Use the **show message disabled** command to list the suppressed messages. All syslog messages are permitted unless explicitly disallowed. The "PIX Startup begin" message cannot be blocked and neither can more than one message per command statement.

monitor	Specify that syslog messages appear on Telnet sessions to the PIX Firewall console.
on	Start sending syslog messages to all output locations. Stop all logging with the **no logging on** command.
port	The port from which the PIX Firewall sends either UDP or TCP syslog messages. This must be same port at which the syslog server listens. For the UDP port, the default is 514 and the allowable range for changing the value is 1025 through 65535. For the TCP port, the default is 1470, and the allowable range is 1025 through 65535. TCP ports only work with the PIX Firewall Syslog Server.
protocol	The protocol over which the syslog message is sent; either **tcp** or **udp**. PIX Firewall only sends TCP syslog messages to the PIX Firewall Syslog Server. You can only view the port and protocol values you previously entered by using the **write terminal** command and finding the command in the listing—the TCP protocol is listed as 6 and the UDP protocol is listed as 17.
queue *queue_size*	Specifies the size of the queue for storing syslog messages. Use this parameter before the syslog messages are processed. The queue parameter defaults to 512 messages, 0 (zero) indicates unlimited (subject to available block memory), and the minimum is one message.
standby	Let the failover standby unit also send syslog messages. This option is disabled by default. You can enable it to ensure that the standby unit's syslog messages stay synchronized should failover occur. However, this option causes twice as much traffic on the syslog server. Disable with the **no logging standby** command.
syslog_id	Specify a message number to disallow or allow. If a message is listed in syslog as %PIX-1-101001, use "101001" as the *syslog_id*. Refer to *Cisco PIX Firewall System Log Messages* for message numbers.
text	Specifies the text string to uniquely identify the syslog messages from the PIX Firewall. The maximum length is 16 characters with no whitespace (blanks) allowed.
timestamp	Specify that syslog messages sent to the syslog server should have a time stamp value on each message.
trap	Set logging level only for syslog messages.

Defaults

EMBLEM format logging is disabled by default.

The **logging device-id** command is disabled by default.

Console logging (the **logging console** command) is disabled by default.

Command Modes

Configuration mode.

Usage Guidelines

The **logging** command lets you enable or disable sending informational messages to the console, to a syslog server, or to an SNMP management station.

The PIX Firewall provides more information in messages sent to a syslog server than at the console, but the console provides enough information to permit effective troubleshooting.

Note Do not use the **logging console** command when the PIX Firewall is in production mode because it degrades system performance. Instead, use the **logging buffered** command to start logging, the **show logging** command to view the messages, and the **clear logging** command to clear the buffer to make viewing the most current messages easier.

The **aaa accounting authentication enable console** command causes syslog messages to be sent (at syslog level 4) each time the configuration is changed from the serial console.

The **show logging** command displays which logging options are enabled. If the **logging buffered** command is in use, the **show logging** command lists the current message buffer. The **show logging disabled** command displays suppressed syslog messages.

logging device-id

The **logging device-id** command displays a unique device ID in non-EMBLEM format syslog messages that are sent to the syslog server. This command is available in PIX Firewall software Version 6.2.2.115 and higher.

If enabled, the PIX Firewall displays the device ID in all non-EMBLEM-formatted syslog messages. However, it does not affect the syslog message text that is in EMBLEM format.

Note The device ID part of the syslog message is viewed through the syslog server only and not directly on the firewall.

If the **ipaddress** option is used, the device ID becomes the specified PIX Firewall interface IP address, regardless of the interface from which the message is sent. This provides a single consistent device ID for all messages sent from the device.

logging history

Set the SNMP message level with the **logging history** command.

logging host

The **logging host** *ip_address* **format emblem** command enables EMBLEM format logging on a per-syslog-server basis. EMBLEM format logging is available for UDP syslog messages only (because the RME syslog analyzer only supports UDP syslog messages). If EMBLEM format logging is enabled for a particular syslog host, then EMBLEM format messages are sent to that host. If the **logging timestamp** option is also enabled, then EMBLEM format messages with a time stamp are sent. EMBLEM format logging is disabled by default.

logging message

To change the level of a syslog message, use the **logging message** *syslog_id* **level** *level* command. The **no logging message** command cannot block the "`%PIX-6-199002: PIX startup completed. Beginning operation`." syslog message.

logging queue

The **logging queue** command lets you specify the size of the syslog message queue for the messages waiting to be processed. When traffic is heavy, messages may be discarded.

The **show logging queue** command lists:

- Number of messages in the queue
- Highest number of messages recorded in the queue

- Number of messages discarded because block memory was not available to process them

logging standby

The **logging standby** command lets the failover standby unit send syslog messages. This option is disabled by default. You can enable it to ensure that the standby unit's syslog messages stay synchronized should failover occur. However, this option causes twice as much traffic on the syslog server. Disable with the **no logging standby** command.

logging timestamp

The **logging timestamp** command requires that the **clock** command be set.

logging trap

Set the syslog message level with the **logging trap** command.

Troubleshooting

If you are using TCP as the logging transport protocol, the PIX Firewall stops passing traffic as a security measure if any of the following error conditions occur: the PIX Firewall is unable to reach the syslog server; the syslog server is misconfigured (such as with PFSS, for example); or the disk is full. (UDP-based logging does not prevent the PIX Firewall from passing traffic if the syslog server fails.)

To enable the PIX Firewall to pass traffic again, do the following:

Step 1 Identify and correct the syslog server connectivity, misconfiguration, or disk space error condition.

Step 2 Enter the command **`logging host inside 10.1.1.1 tcp/1468`** to enable the logging again.

Alternately, you can change the logging to default logging on UDP/514 by issuing the command **`logging host inside 10.1.1.1`**. UDP-based logging passes traffic even if the syslog server fails.

For more information

For more information on syslog and the use of the **logging** command, refer to *Cisco PIX Firewall System Log Messages*. You can also use *Cisco PIX Firewall System Log Messages* to get the message numbers that can be individually suppressed with the **logging message** command.

Examples

The following example shows how to start console logging and view the results:

```
pixfirewall(config)# logging buffered debugging
pixfirewall(config)# show logging
Syslog logging: enabled
    Timestamp logging: disabled
    Console logging: disabled
    Monitor logging: disabled
    Buffer logging: level debugging, 37 messages logged
    Trap logging: disabled
305001: Portmapped translation built for gaddr 209.165.201.5/0 laddr 192.168.1.2/256
...
```

The line of output starting with 305001 shows a translation to a PAT global through global address 209.165.201.5 from a host at 192.168.1.2. The "305001" identifies a syslog message for creating a translation through a PAT global. Refer to *Cisco PIX Firewall System Log Messages* for more information on syslog messages.

The following is sample output from the **show logging** command with the **logging device-id hostname** command configured on a host named **pixfirewall-1** (notice the last line):

```
pixfirewall-1(config)# logging device-id hostname
pixfirewall-1(config)# show logging
Syslog logging: disabled
Facility: 20
Timestamp logging: disabled
Standby logging: disabled
Console logging: level debugging, 0 messages logged
Monitor logging: level debugging, 0 messages logged
Buffer logging: disabled
Trap logging: disabled
History logging: disabled
Device ID: hostname "pixfirewall-1"
```

The next example lists the output of the **logging queue** and **show logging queue** commands:

```
pixfirewall(config)# logging queue 0
pixfirewall(config)# show logging queue
Logging Queue length limit : Unlimited
Current 5 msg on queue, 3513 msgs most on queue, 1 msg discard.
```

In this example, the **logging queue** command is set to 0, which means you want an unlimited number of messages; in other words, all syslog messages, to be processed. The **show logging queue** command shows that 5 messages are queued, 3513 messages was the greatest number of messages in the queue at one time since the PIX Firewall was last booted, and that 1 message was discarded. Even though set for unlimited, should the amount of block memory be exhausted, messages can still be discarded.

The following is sample output from the **show logging** command output when the TCP syslog server is unreachable. Consequently, the PIX Firewall stops passing traffic and logging to the inside is set as **disabled**:

```
pixfirewall(config)# show logging
Syslog logging: enabled
Timestamp logging: enabled
Standby logging: disabled
Console logging: disabled
Monitor logging: disabled
Buffer logging: level debugging, 827 messages logged
Trap logging: level debugging, facility 20, 840 messages logged
   Logging to inside 10.1.1.1 tcp/1468 disabled
```

The following examples show how to change the level of a syslog message and display its current and default level:

```
pixfirewall(config)# logging message 403503
pixfirewall(config)# show logging message 403503
syslog 403503: default-level errors (enabled)

pixfirewall(config)# logging message 403503 level 1
pixfirewall(config)# show logging message 403503
syslog 403503: default-level errors, current-level alerts (enabled)

pixfirewall(config)# logging message 403503 level 6
pixfirewall(config)# show logging message 403503
syslog 403503: default-level errors, current-level informational (enabled)

pixfirewall(config)# logging message 403503 level 3
pixfirewall(config)# show logging message 403503
syslog 403503: default-level errors (enabled)
```

Related Commands

auto-update	Configures auto update support.
telnet	Adds Telnet access to the firewall console and sets the idle timeout.
terminal	Sets terminal line parameters.

login

To log into privileged mode using the local user database (see the **username** command) or to change user names, use the **login** command in unprivileged mode.

login

Syntax Description

login	Logs in as a particular user.

Command Modes

Unprivileged mode.

Usage Guidelines

From unprivileged mode, you can log in to privileged mode as any username in the local database using the **login** command. The **login** command is similar to the **enable** command when you have **enable** authentication turned on (see the **aaa authentication console** command). Unlike **enable** authentication, the **login** command can only use the local username database, and authentication is always required with this command. You can also change users using the **login** command from any CLI mode.

To allow users to access privileged mode (and all commands) when they log in, set the user privilege level to 2 (the default) through 15. If you configure local command authorization, then the user can only enter commands assigned to that privilege level or lower. See the **aaa authorization command** for more information.

Caution If you add users to the local database who can gain access to the CLI and whom you do not want to enter privileged mode, you should configure command authorization. Without command authorization, users can access privileged mode (and all commands) at the CLI using their own password if their privilege level is 2 or greater (2 is the default). Alternatively, you can use RADIUS or TACACS+ authentication, or you can set all local users to level 1 so you can control who can use the system enable password to access privileged mode.

Examples

The following example shows the prompt after you enter the **login** command:

```
pixfirewall> login
Username:
```

Related Commands

privilege	Configures privilege levels for commands.
username	Configures the local user authentication database.