



## M through R Commands

---

### mtu

Specify the maximum transmission unit ( MTU) for an interface. (Configuration mode.)

```
mtu if_name bytes  
no mtu [if_name bytes]  
show mtu
```

---

#### Syntax Description

<i>if_name</i>	The internal or external network interface name.
<i>bytes</i>	The number of bytes in the MTU, in the range of 64 to 65,535 bytes. The value specified depends on the type of network connected to the interface.

---

---

#### Usage Guidelines

The **mtu** command sets the size of data sent on a connection. Data larger than the MTU value is fragmented before being sent. The minimum value for *bytes* is 64 and the maximum is 65,535 bytes.

PIX Firewall supports the IP Path MTU Discovery mechanism, as defined in RFC 1191. IP Path MTU Discovery allows a host to dynamically discover and cope with differences in the maximum allowable maximum transmission unit (MTU) size of the various links along the path. Sometimes a PIX Firewall is unable to forward a datagram because it requires fragmentation (the packet is larger than the MTU you set for the interface), but the “don't fragment” (DF) bit is set. The network software sends a message to the sending host, alerting it to the problem. The host will have to fragment packets for the destination so that they fit the smallest packet size of all the links along the path.

For Ethernet interfaces, the default MTU is 1500 bytes in a block, which is also the maximum. This value is sufficient for most applications, but you can pick a lower number if network conditions warrant it.

For Token Ring and FDDI, the default is 8192 bytes.

The **no mtu** command resets the MTU block size to 1500 for Ethernet interfaces and 8192 for Token Ring. The **show mtu** command displays the current block size. The **show interface** command also shows the MTU value.

---

**Examples**

The following example shows use of the **mtu** command for use with Token Ring and Ethernet:

```
interface token-ring0 16mbps
interface ethernet0 auto
mtu inside 8192
show mtu
mtu outside 1500
mtu inside 8192
```

# name/names

Associate a name with an IP address. (Configuration mode.)

**name** *ip\_address name*

**no name** [*ip\_address name*]

**names**

**no names**

**clear names**

**show names**

## Syntax Description

<i>ip_address</i>	The IP address of the host being named.
<i>name</i>	The name assigned to the IP address. Allowable characters are <b>a</b> to <b>z</b> , <b>A</b> to <b>Z</b> , <b>0</b> to <b>9</b> , a dash, and an underscore. The <i>name</i> cannot start with a number. If the name is over 16 characters long, the <b>name</b> command fails.

## Usage Guidelines

Use the **name** command to identify a host by a text name. The names you define become like a host table local to the PIX Firewall. Because there is no connection to DNS or `/etc/hosts` on UNIX servers, use of this command is a mixed blessing—it makes configurations much more readable but introduces another level of abstraction to administer; not only do you have to add and delete IP addresses to your configuration as you do now, but with this command, you need to ensure that the host names either match existing names or you have a map to list the differences.

The **names** command enables use of the **name** command to map text strings to IP addresses. The **clear names** and **no names** commands are the same and disable use of the **name** text strings. The **show names** command lists the **name** command statements in the configuration.

## Notes

1. You must first use the **names** command before using the **name** command. Use the **name** command immediately after the **names** command and before you use the write memory command.
2. To disable displaying **name** values, use the **no names** command.
3. Only one name can be associated with an IP address.
4. Both the **name** and **names** command statements are saved in the configuration.
5. While the **name** command will let you assign a name to a network mask, no other PIX Firewall command requiring a mask will let you use the name as a mask value. For example, the following command is accepted.

```
name 255.255.255.0 class-C-mask
```



## Note

None of the commands in which a mask is required can process the “class-C-mask” as an accepted network mask.

---

**Examples**

In the example that follows, the **names** command enables use of the **name** command. The **name** command substitutes **pix\_inside** for references to 192.168.42.3, and **pix\_outside** for 209.165.201.3. The **ip address** commands use these names while assigning IP addresses to the network interfaces. The **no names** command disables the **name** command values from displaying. Subsequent use of the **names** command restores their display.

```
names
name 192.168.42.3 pix_inside
name 209.165.201.3 pix_outside
ip address inside pix_inside 255.255.255.0
ip address outside pix_outside 255.255.255.224
show ip address
inside ip address pix_inside mask 255.255.255.0
outside ip address pix_outside mask 255.255.255.224
no names
show ip address
inside ip address 192.168.42.3 mask 255.255.255.0
outside ip address 209.165.201.3 mask 255.255.255.224
names
show ip address
inside ip address pix_inside mask 255.255.255.0
outside ip address pix_outside mask 255.255.255.224
```

# nameif

Name interfaces and assign security level. (Configuration mode.)

**nameif** *hardware\_id* *if\_name* *security\_level*

**show nameif**

**clear nameif**

## Syntax Description

<i>hardware_id</i>	<p>The hardware name for the network interface that specifies the interface's slot location on the PIX Firewall motherboard. Interface boards are numbered from the leftmost slot nearest the power supply as slot 0. The internal network interface must be in slot 1. The lowest <i>security_level</i> external interface board is in slot 0 and the next lowest <i>security_level</i> external interface board is in slot 2.</p> <p>Possible choices are <b>ethernet<math>n</math></b> for Ethernet or <b>token-ring<math>n</math></b> for Token Ring. These names can be abbreviated with any leading characters in the name; for example, <b>ether1</b>, <b>e2</b>, <b>token0</b>, or <b>t0</b>.</p>
<i>if_name</i>	<p>A name for the internal or external network interface of up to 48 characters in length. This name can be uppercase or lowercase. By default, PIX Firewall names the inside interface "inside," the outside interface "outside," and any perimeter interface "intfn" where <math>n</math> is 2 through 5.</p>
<i>security_level</i>	<p>Either <b>0</b> for the outside network or <b>100</b> for the inside network. Perimeter interfaces can use any number between <b>1</b> and <b>99</b>. By default, PIX Firewall sets the security level for the inside interface to <b>security100</b> and the outside interface to <b>security0</b>. The first perimeter interface is initially set to <b>security10</b>, the second to <b>security15</b>, the third to <b>security20</b>, and the fourth perimeter interface to <b>security25</b> (a total of 6 interfaces are permitted, with a total of 4 perimeter interfaces permitted).</p> <p>For access from a higher security to a lower security level, <b>nat</b> and <b>global</b> commands or <b>static</b> commands must be present. For access from a lower security level to a higher security level, <b>static</b> and <b>access-list</b> commands must be present.</p> <p>Interfaces with the same security level cannot communicate with each other. We recommend that every interface have a unique security level.</p>

## Usage Guidelines

The **nameif** command allows you to assign a name to an interface. You can use this command to assign interface names if you have more than two network interface circuit boards in your PIX Firewall. The first two interfaces have the default names **inside** and **outside**. The **inside** interface has a default security level of 100, the **outside** interface has a default security level of 0. The **clear nameif** command reverts **nameif** command statements to default interface names and security levels.

## Usage Notes

1. If you change the *hardware\_id* of the outside interface; for example, from ethernet0 to ethernet1, PIX Firewall changes every reference to the outside interface in your configuration to inside, which can cause problems with **route**, **ip**, and other command statements that affect the flow of traffic through the PIX Firewall.
2. After changing a **nameif** command, use the **clear xlate** command.

3. The inside interface cannot be renamed or given a different security level. The outside interface can be renamed, but not given a different security level.
4. An interface is always “external” with respect to another interface that has a higher security level.

---

**Examples**

The following example shows use of the **nameif** command:

```
nameif ethernet2 perimeter1 sec50
nameif ethernet3 perimeter2 sec20
```

---

**Related Commands**

- [interface](#)

# nat

Associate a network with a pool of global IP addresses. (Configuration mode.)

```
nat [(if_name)] nat_id local_ip [netmask [max_conns [em_limit]]] [norandomseq]
```

```
nat [(if_name)] 0 access-list acl_name
```

```
nat [(if_name)] 0 local_ip [netmask [max_conns [em_limit]]] [norandomseq]
```

```
no nat [(if_name)] nat_id local_ip [netmask [max_conns [em_limit]]] [norandomseq]
```

```
no nat [(if_name)] 0 access-list acl_name
```

```
show nat
```

```
clear nat
```

## Syntax Description

<i>if_name</i>	The internal network interface name.  If the interface is associated with an access list, then the <i>if_name</i> is the higher security level interface name.
<i>nat_id</i>	All <b>nat</b> command statements with the same <i>nat_id</i> are in the same <b>nat</b> group. Use the <i>nat_id</i> in the <b>global</b> command statement; for example:  <pre><b>nat</b> (perimeter) 1 0 0 <b>global</b> (outside) 1 209.165.201.1-209.165.201.30 <b>netmask</b> 255.255.255.224</pre> This example associates the <b>nat</b> command with the <b>global</b> command via the <i>nat_id</i> .  The <i>nat_id</i> is an arbitrary positive number between 0 and two billion. This number can be the same as the ID used with the <b>outbound</b> and <b>apply</b> commands.  Specify <b>0</b> with IP addresses and netmasks to identify internal networks that desire only outbound identity address translation. Specify <b>0</b> with the <b>access-list</b> option to specify traffic that should be exempted from NAT.
<b>access-list</b>	Associate an <b>access-list</b> command statements to the <b>nat 0</b> command.
<i>local_ip</i>	Internal network IP address to be translated. You can use <b>0.0.0.0</b> to allow all hosts to start outbound connections. The <b>0.0.0.0</b> <i>local_ip</i> can be abbreviated as <b>0</b> .
<i>netmask</i>	Network mask for <i>local_ip</i> . You can use <b>0.0.0.0</b> to allow all outbound connections to translate with IP addresses from the global pool.
<i>max_conns</i>	The maximum TCP connections permitted from the interface you specify.
<b>clear</b>	Removes <b>nat</b> command statements from the configuration.
<i>em_limit</i>	The embryonic connection limit. The default is 0, which means unlimited connections. Set it lower for slower systems, higher for faster systems.
<b>norandomseq</b>	Do not randomize the TCP packet's sequence number. Only use this option if another inline firewall is also randomizing sequence numbers and the result is scrambling the data. Use of this option opens a security hole in the PIX Firewall.

**Usage Guidelines**

The **nat** command lets you enable or disable address translation for one or more internal addresses. Address translation means that when a host starts an outbound connection, the IP addresses in the internal network are translated into global addresses. Network Address Translation (NAT) allows your network to have any IP addressing scheme and the PIX Firewall protects these addresses from visibility on the external network.

**Note**

If not explicitly included in the **nat** command, the PIX Firewall derives the network mask from the class of the IP address. For example, the command **nat 0 10.130.36.0** causes all addresses in the 10.0.0.0 network to be translated and not only those in the 10.130.36.0 network. For this reason, you should specify the network mask when configuring an IP address that is not classful.

The **nat (if\_name) 0 access-list acl\_name** command lets you exempt traffic that is matched by the **access-list** command statements from the NAT services. Adaptive Security remains in effect with the **nat 0 access-list** command. The extent to which the inside hosts are accessible from the outside depends on the **access-list** command statements that permit inbound access. The *if\_name* is the higher security level interface name. The *acl\_name* is the name you use to identify the **access-list** command statement.

With PIX Firewall software version 5.3 and later, there is no longer a restriction on having the **nat 0** command (Identity NAT) and the **nat 0 access-list** command configured at the same time. Both the **nat 0** command and the **nat 0 access-list** command may be configured concurrently.

The new **access-list** option changes the behavior of the **nat 0** command. (Without the **access-list** option, the command is backward compatible with previous versions.)

The **nat 0** command implemented the identity feature; this new version of the command disables NAT. Specifically, the new behavior disables proxy ARPing for the IP addresses in the **nat 0** command statement.

**Note**

The access list you specify with the **nat 0 access-list** command will not work with an **access-list** command statement that contains a port specification. The following sample command statements will not work.

```
access-list no-nat permit tcp host xx.xx.xx.xx host yy.yy.yy.yy
nat (inside) 0 access-list no-nat
```

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

The connection limit allows you to set the maximum number of outbound connections that can be started with the IP address criteria you specify. The embryonic connection limit allows you to prevent a type of attack where processes are started without being completed. An embryonic connection is a connection that someone attempted but has not completed and has not yet seen data. Every connection is embryonic until it sets up.

You can use the **no nat** command to remove a **nat** command statement and you can use the **show nat** command to view **nat** command statements in the current configuration.

Table 7-1 helps you decide when to use the **nat** or **static** commands for access between the various interfaces in the PIX Firewall. For this table, assume that the security levels are 40 for dmz1 and 60 for dmz2.

**Table 7-1 Interface Access Commands by Interface**

From This Interface	To This Interface	Use This Command		From This Interface	To This Interface	Use This Command
inside	outside	<b>nat</b>		dmz2	outside	<b>nat</b>
inside	dmz1	<b>nat</b>		dmz2	dmz1	<b>nat</b>
inside	dmz2	<b>nat</b>		dmz2	inside	<b>static</b>
dmz1	outside	<b>nat</b>		outside	dmz1	<b>static</b>
dmz1	dmz2	<b>static</b>		outside	dmz2	<b>static</b>
dmz1	inside	<b>static</b>		outside	inside	<b>static</b>

The rule of thumb is that for access from a higher security level interface to a lower security level interface, use the **nat** command. From lower security level interface to a higher security level interface, use the **static** command.

#### Usage Notes

1. You can enable identity address translation with the **nat 0** command. Use this command when you have IP addresses that are the same as those used on more than one interface. Adaptive Security remains in effect with the **nat 0** command. The extent to which the inside hosts are accessible from the outside depends on the **access-list** command statements that permit inbound access.

Addresses on each interface must be on a different subnet. See Appendix D “TCP/IP Reference Information” of the *Cisco PIX Firewall and VPN Configuration Guide* for more information about subnetting.

The **nat 0 10.2.3.0** command means let those IP addresses in the 10.2.3.0 net appear on the outside without translation. All other hosts are translated depending on how their **nat** command statements appear in the configuration.

2. The **nat 1 0 0** command means that all outbound connections can pass through the PIX Firewall with address translation. If you use the **nat (inside) 1 0 0** command, users can start connections on any interface with a lower security level, on the both perimeter interfaces and the outside interface. With NAT in effect, you must also use the **global** command statement to provide a pool of addresses through which translated connections pass. In effect, you use the **nat** command statement to specify from which interface connections can originate and you use the **global** command statement to determine at which interface connections can occur. The NAT ID must be the same on the **nat** and **global** command statements.
3. The **nat 1 10.2.3.0** command means that only outbound connections originating from the inside host 10.2.3.0 can pass through the PIX Firewall to go to their destinations with address translation.

#### Examples

The following example specifies with **nat** command statements that all the hosts on the 10.0.0.0 and 10.3.3.0 inside networks can start outbound connections. The **global** command statements create a pool of global addresses as follows:

```
nat (inside) 1 10.0.0.0 255.0.0.0
global (outside) 1 209.165.201.24-209.165.201.27 netmask 255.255.255.224
```

```
global (outside) 1 209.165.201.30
nat (inside) 3 10.3.3.0 255.255.255.0
global (outside) 3 209.165.201.10-209.165.201.24 netmask 255.255.255.224
```

### Related Commands

- [global](#)
- [outbound/apply](#)

When using the **nat 0** command, if you want the addresses to be visible from the outside network, use **static** and **access-list** command statements:

```
nat (inside) 0 209.165.201.0 255.255.255.224
static (inside, outside) 209.165.201.0 209.165.201.0 netmask 255.255.255.224
access-list acl_out permit host 10.0.0.1 209.165.201.0 255.255.255.224 eq ftp
access-group acl_out in interface outside
```

```
nat (inside) 0 209.165.202.128 255.255.255.224
static (inside, outside) 209.165.202.128 209.165.202.128 netmask 255.255.255.224
access-list acl_out permit tcp host 10.0.0.1 209.165.202.128 255.255.255.224 eq ftp
access-group acl_out in interface outside
```

...

The following example shows use of the **nat 0 access-list** command to permit internal host 10.1.1.15, accessible through the inside interface, “inside,” to bypass NAT when connecting to outside host 10.2.1.3.

```
access-list no-nat permit ip host 10.1.1.15 host 10.2.1.3
nat (inside) 0 access-list no-nat
```

The following commands will disable all NAT on a PIX Firewall with three interfaces:

```
access-list all-ip-packet permit ip 0 0 0 0
nat (dmz) 0 access-list all-ip-packet
nat (inside) 0 access-list all-ip-packet
```

# outbound/apply

Create an access list for controlling Internet use. (Configuration mode.)

```
outbound list_ID permit | deny ip_address [netmask [port[-port]] [protocol]
```

```
outbound list_ID except ip_address [netmask [port[-port]] [protocol]
```

```
clear outbound
```

```
no outbound [list_ID permit | deny ip_address [netmask [port[-port]] [protocol]]
```

```
no outbound [list_ID except ip_address [netmask [port[-port]] [protocol]]
```

```
show outbound
```

```
apply [(if_name)] list_ID outgoing_src | outgoing_dest
```

```
clear apply
```

```
no apply [(if_name)] list_ID outgoing_src | outgoing_dest
```

```
show apply [(if_name)] [list_ID outgoing_src | outgoing_dest]
```

## Syntax Description

<b>outbound</b>	The <b>outbound</b> command, in conjunction with the <b>apply</b> command, uses access lists to control a filtering function on outgoing packets from the PIX Firewall. The filters can be based on the source IP address, the destination IP address, and the destination port/protocol as specified by the rules.  The use of an <b>outbound</b> command requires use of the <b>apply</b> command. The <b>apply</b> command allows you to specify whether the access control list applies to inside users' ability to start outbound connections with the <b>apply</b> command's <b>outgoing_src</b> option, or whether the access list applies to inside users' ability to access servers on the outside network with the <b>apply</b> command's <b>outgoing_dest</b> option.  For more information, see "Outbound List Rules" and the <b>access-list</b> command. The <b>outbound</b> command has been superseded by the <b>access-list</b> command.
<i>list_ID</i>	A tag number for the access list. The access list number you use must be the same for the <b>apply</b> and <b>outbound</b> commands. This value must be a positive number from 1 to 1599. This number can be the same as what you use with the <b>nat</b> and <b>global</b> commands. This number is just an arbitrary number that groups <b>outbound</b> command statements to an <b>apply</b> command statement. <i>List_IDs</i> are processed sequentially in descending order.  For more information, see " <a href="#">Outbound List Rules</a> ."
<b>no</b> <b>outbound</b>	Removes a single <b>outbound</b> command statement from the configuration.
<b>clear</b> <b>outbound</b>	Removes all <b>outbound</b> command statements from the configuration.
<b>show</b> <b>outbound</b>	Displays the <b>outbound</b> command statements in the configuration.
<b>permit</b>	Allow the access list to access the specified IP address and port.
<b>deny</b>	Deny the access list access to the specified IP address and port.

<b>except</b>	<p>Create an exception to a previous <b>outbound</b> command. An <b>except</b> command statement applies to <b>permit</b> or <b>deny</b> command statements only with the same access list ID.</p> <p>When used with <b>apply outgoing_src</b>, the IP address of an <b>except</b> command statement applies to the destination address.</p> <p>When used with <b>apply outgoing_dest</b>, the IP address of an <b>except</b> command statement applies to the source address.</p> <p>See “<a href="#">Outbound List Rules</a>” for more information.</p>
<i>ip_address</i>	The IP address for this access list entry. Do not specify a range of addresses. The 0.0.0.0 <i>ip_address</i> can be abbreviated as 0.
<i>netmask</i>	The network mask for comparing with the IP address; 255.255.255.0 causes the access list to apply to an entire Class C address. 0.0.0.0 indicates all access. The 0.0.0.0 <i>netmask</i> can be abbreviated as 0.
<i>port</i>	A port or range of ports that the access list is permitted or denied access to. See the “ <a href="#">Ports</a> ” section in <a href="#">Chapter 1, “Using PIX Firewall Commands”</a> for a list of valid port literal names.
<i>protocol</i>	Limit outbound access to <b>udp</b> , <b>tcp</b> , or <b>icmp</b> protocols. If a protocol is not specified, the default is <b>tcp</b> .
<i>if_name</i>	The network interface originating the connection.
<b>outgoing_src</b>	Deny or permit an internal IP address the ability to start outbound connections using the service(s) specified in the <b>outbound</b> command.
<b>outgoing_dest</b>	Deny or permit access to an external IP address using the service(s) specified in the <b>outbound</b> command.
<b>apply</b>	Specifies whether the access control list applies to inside users’ ability to start outbound connections with <b>apply</b> command’s <b>outgoing_src</b> option, or whether the access list applies to inside users’ ability to access servers on the outside network with the <b>apply</b> command’s <b>outgoing_dest</b> option.
<b>noapply</b>	Removes a single <b>apply</b> command statement from the configuration.
<b>clear apply</b>	Removes all the <b>apply</b> command statements from the configuration. The <b>show apply</b> command displays the <b>apply</b> command statements in the configuration.
<b>show apply</b>	Displays the <b>apply</b> command statements in the configuration.

### Usage Guidelines

The **outbound** command creates an access list that allows you to specify the following:

- Whether inside users can create outbound connections
- Whether inside users can access specific outside servers
- What services inside users can use for outbound connections and for accessing outside servers
- Whether outbound connections can execute Java applets on the inside network

Outbound lists are filters on outgoing packets from the PIX Firewall. The filter can be based on the source IP address, the destination IP address, and the destination port/protocol as specified by the rules. The use of an **outbound** command requires use of the **apply** command. The **apply** command allows you to specify whether the access control list applies to inside users’ ability to start outbound connections with **apply** command’s **outgoing\_src** option, or whether the access list applies to inside users’ ability to access servers on the outside network with the **apply** command’s **outgoing\_dest** option.

**Note**

The **outbound** command has been superseded by the **access-list** command. We recommend that you migrate your **outbound** command statements to **access-list** command statements to maintain future compatibility.

The **java** option has been replaced by the **filter java** command.

After adding, removing, or changing **outbound** command statements, use the **clear xlate** command.

Use the **no outbound** command to remove a single **outbound** command statement from the configuration. Use the **clear outbound** command to remove all **outbound** command statements from the configuration. The **show outbound** command displays the **outbound** command statements in the configuration.

Use the **no apply** command to remove a single **apply** command statement from the configuration. Use the **clear apply** command statement to remove all the **apply** command statements from the configuration. The **show apply** command displays the **apply** command statements in the configuration.

### Outbound List Rules

Rules, written as **outbound list\_ID...** command statements are global to the PIX Firewall, they are activated by **apply list\_ID outgoing\_src | outgoing\_dest** command statements. When applied to *outgoing\_src*, the source IP address, the destination port, and protocol are filtered. When applied to *outgoing\_dest*, the destination IP address, port, and protocol are filtered.

The *outgoing\_src* option and *outgoing\_dest* outbound lists are filtered independently. If any one of the filters contain the **deny** option, the outbound packet is denied. When multiple rules are used to filter the same packet, the best matched rule takes effect. The best match is based on the IP address mask and the port range check. More strict IP address masks and smaller port ranges are considered a better match. If there is a tie, a **permit** option overrides a **deny** option.

Rules are grouped by a *list\_ID*. Within each *list\_ID*, **except** rules (that is, **outbound n except ...**) can be set. The **except** option reverses the best matched rule of **deny** or **permit**. In addition, PIX Firewall filters the specified IP address and mask in the rule for the destination IP address of the outbound packet if the list is applied to the *outbound\_src*. Alternatively, PIX Firewall filters the source IP address if the list is applied to the *outgoing\_dest*. Furthermore, the **except** rules only apply to rules with the same *list\_ID*. A single **except** rule within a *list\_ID* without another **permit** or **deny** rule has no effect. If multiple **except** rules are set, the best match is checked for which **except** to apply.

The **outbound** command rules are now sorted by the best match checking. Use the **show outbound** command to see how the best match is judged by the PIX Firewall.

### Usage Notes

1. If **outbound** commands are not specified, the default behavior is to permit all outbound traffic and services from inside hosts.
2. After adding, changing, or removing an **outbound** and **apply** command statement group, use the **clear xlate** command to make the IP addresses available in the translation table.
3. The **outbound** commands are processed linearly within a *list\_ID*. In addition, *list\_IDs* are processed sequentially in descending order. For example, the first command statement you specify in an **outbound** list is processed first, then the next **outbound** command statement in that list, and so on. Similarly, *list\_ID* 10 is processed before *list\_ID* 20, and so on.

- When using **outbound** commands, it is often helpful to deny or permit access to the many before you deny or permit access to the specific. Start with an interface-wide specification such as the following command that denies all hosts from starting connections.

```
outbound 1 deny 0 0 0
apply (inside) 1 outgoing_src
```

Then add command statements that permit or deny hosts access to specific ports, for example:

```
outbound 1 deny 0 0 0
outbound 1 permit 10.1.1.1 255.255.255.255 23 tcp
outbound 1 permit 10.1.1.1 255.255.255.255 80 tcp
apply (inside) 1 outgoing_src
```

You could state this same example as follows with the **except** option.

```
outbound 1 deny 0 0 0
outbound 1 except 209.165.201.11 255.255.255.255 23 tcp
outbound 1 except 209.165.201.11 255.255.255.255 80 tcp
apply (inside) 1 outgoing_src
```

In the preceding **outbound except** command statement, IP address 209.165.201.11 is the destination IP address, not the source address. This means that everyone is denied outbound access, except those users going to 209.165.201.11 via Telnet (port 23) or HTTP (port 80).

- If you permit access to port 80 (**http**), this also permits Java applets to be downloaded. You must have a specific **deny** command statement to block Java applets.
- The maximum number of **outbound** list entries in a configuration is 1599.
- Outbound lists have no effect on **access-list** command statement groups.
- The use of the **access-group** command statement overrides the **conduit** and **outbound** command statements for the specified interface name.

## Examples

The first **outbound** group sets inside hosts so that they can only see and Telnet to perimeter hosts, and do DNS lookups. In this example, the perimeter network address is 209.165.201.0 and the network mask is 255.255.255.224.

```
outbound 9 deny 0.0.0.0 0.0.0.0 0 0
outbound 9 except 209.165.201.0 255.255.255.224 23 tcp
outbound 9 except 0.0.0.0 0.0.0.0 53 udp
```

The next **outbound** group in this same example lets hosts 10.1.1.11 and 10.1.1.12 go anywhere:

```
outbound 11 deny 0.0.0.0 0.0.0.0 0 0
outbound 11 permit 10.1.1.11 255.255.255.255 0 0
outbound 11 permit 10.1.1.12 255.255.255.255 0 0
outbound 11 permit 0.0.0.0 0.0.0.0 21 tcp
outbound 11 permit 10.3.3.3 255.255.255.255 143 tcp
```

This last **outbound** group in this same example lets hosts on the perimeter only access TCP ports 389 and 30303 and UDP port 53 (DNS). Finally, the **apply** command statements set the **outbound** groups so that the permit and deny rules affect access to all external addresses.

```
outbound 13 deny 0.0.0.0 0.0.0.0 0 0
outbound 13 permit 0.0.0.0 0.0.0.0 389 tcp
outbound 13 permit 0.0.0.0 0.0.0.0 30303 tcp
outbound 13 permit 0.0.0.0 0.0.0.0 53 udp

apply (inside) 9 outgoing_src
apply (inside) 11 outgoing_src
apply (perim) 13 outgoing_src
```

### Controlling Outbound Connections

The following example prevents all inside hosts from starting outbound connections:

```
outbound 1 deny 0 0 0
apply (inside) 1 outgoing_src
```

The **0 0 0** at the end of the command means all IP addresses (**0** is the same as **0.0.0.0**), with a 0.0.0.0 subnet mask and for all services (port value is zero).

Conversely, the following example permits all inside hosts to start connections to the outside (this is the default if an access list is not created):

```
outbound 1 permit 0 0 0
apply (inside) 1 outgoing_src
```

### Controlling Inside Hosts' Access to Outbound Services

The following example prevents inside host 192.168.1.49 from accessing the World Wide Web (port 80):

```
outbound 11 deny 192.168.1.49 255.255.255.255 80 tcp
apply (inside) 11 outgoing_src
```

### Controlling Inside Hosts' Access to Outside Servers

If your employees are spending too much time examining GIF images on a particular website with two web servers, you can use the following example to restrict this access:

```
outbound 12 deny 192.168.146.201 255.255.255.255 80 tcp
outbound 12 deny 192.168.146.202 255.255.255.255 80 tcp
apply (inside) 12 outgoing_dest
```

### Using except Command Statements

An **except** command statement only provides exception to items with the same *list\_ID*. Consider the following example.

```
outbound 9 deny 0.0.0.0 0.0.0.0 0 0
outbound 9 except 10.100.0.0 255.255.0.0 23 tcp
outbound 9 except 0.0.0.0 0.0.0.0 53 udp
outbound 11 deny 0.0.0.0 0.0.0.0 0 0
outbound 11 permit 10.1.1.11 255.255.255.255 0 0
outbound 11 permit 10.1.1.12 255.255.255.255 0 0
outbound 11 permit 0.0.0.0 0.0.0.0 21 tcp
outbound 11 permit 10.3.3.3 255.255.255.255 143 tcp
outbound 13 deny 0.0.0.0 0.0.0.0 0 0
outbound 13 permit 0.0.0.0 0.0.0.0 389 tcp
outbound 13 permit 0.0.0.0 0.0.0.0 30303 tcp
outbound 13 permit 0.0.0.0 0.0.0.0 53 udp
```

In the preceding examples, the following two command statements work against other command statements in list 9 but not in lists 11 and 13:

```
outbound 9 except 10.100.0.0 255.255.0.0 23 tcp
outbound 9 except 0.0.0.0 0.0.0.0 53 udp
```

In the following example, the set of **deny**, **permit**, and **except** option command statements denies everybody from connecting to external hosts except for DNS queries and Telnet connections to hosts on 10.100.0.0. The host with IP address 10.1.1.11 is permitted outbound access, and has access to everywhere *except* to 10.100.0.0 via Telnet and anywhere to use DNS.

```
outbound 1 deny 0.0.0.0 0.0.0.0 0 tcp
outbound 1 permit 10.1.1.11 255.255.255.255 0 tcp
outbound 1 except 10.100.0.0 255.255.0.0 23 tcp
outbound 1 except 0.0.0.0 0.0.0.0 53 udp
apply (inside) outgoing_src
```

# pager

Enable or disable screen paging. (Privileged mode.)

**pager** [*lines number*]

**clear pager**

**no pager**

**show pager**

## Syntax Description

<i>number</i>	The number of lines before the More prompt appears. The minimum is <b>1</b> . Use <b>0</b> to disable paging.
---------------	---

## Usage Guidelines

The **pager lines** command allows you to specify the number of lines in a page before the More prompt appears. The **pager** command enables display paging, and **no pager** disables paging and lets output display completely without interruption. If you set **pager lines** to some value and want to revert back to the default, enter the **pager** command without options. The **clear pager** command resets the number of lines in a page to 24.

Use **pager 0** to disable paging.

The **show pager** command displays **pager** status.

When paging is enabled, the following prompt appears:

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

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

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

To return to the command line, press the **q** key.

## Examples

The following example shows use of the **pager** command:

```
pixfirewall# pager lines 2
pixfirewall# ping inside 10.0.0.42
    10.0.0.42 NO response received -- 1010ms
    10.0.0.42 NO response received -- 1000ms
<--- More --->
```

# passwd

Set password for Telnet access to the PIX Firewall console. (Privileged mode.)

**passwd** *password* [**encrypted**]

**clear** passwd

**show** passwd

## Syntax Description

<i>password</i>	A case-sensitive password of up to 16 alphanumeric and special characters. Any character can be used in the password except a question mark and a space.
<b>encrypted</b>	Specifies that the password you entered is already encrypted. The <i>password</i> you specify with the <b>encrypted</b> option must be 16 characters in length.

## Usage Guidelines

The **passwd** command sets a password for Telnet access to the PIX Firewall console. An empty password is also changed into an encrypted string. However, any use of a **write** command displays or writes the passwords in encrypted form. Once passwords are encrypted, they are not reversible back to plain text. The **clear passwd** command resets the password to “cisco.”



### Note

Write down the new password and store it in a manner consistent with your site’s security policy. Once you change this password, you cannot view it again.

## Examples

The following example shows use of the **passwd** command:

```
passwd watag00s1am
show passwd
passwd jMorNbK0514fadBh encrypted
```

## Related Commands

- [enable password](#)

# pdm

A new family of commands support Cisco PIX Device Manager (PDM) communication with a PIX Firewall over an HTTP server. The **pdm disconnect** command allows you to disconnect a specific PDM session using a *session\_id* obtained with the **show pdm sessions** command. The **show pdm sessions** command lists all the open PDM sessions going to a PIX Firewall. (Configuration mode.)



## Note

The **pdm disconnect** command and the **show pdm sessions** command are accessible through the command line. The **clear pdm**, **pdm history commands**, **pdm location**, and **pdm logging** commands may appear in your configuration and are available through the CLI, but they are designed to work as internal PDM-to-PIX Firewall commands accessible through PDM.

**clear pdm**

**pdm disconnect** *session\_id*  
**show pdm sessions**

**[no] pdm history enable**

**show pdm history** [**view** {**all**|**12h**|**5d**|**60m**|**10m**}][**snapshot**] [**feature**  
{**all**|**blocks**|**cpu**|**failover**|**ids**|**interface** <*if\_name*>|**memory**|**perfmon**|**xlates**}][**pdmclient**]

**pdm location** *ip\_address netmask if\_name*

**pdm logging** [*level* [*messages*]]  
**no pdm logging**  
**show pdm logging**

## Syntax Description

<b>pdm</b>	Pertaining to the Cisco PIX Device Manager.
<b>clear pdm</b>	Removes all locations, disables logging and clears the PDM buffer. Internal PDM command.
<b>pdm disconnect</b>	Disconnects the specified PDM session from the PIX Firewall.
<i>session_id</i>	PDM session ID number available from the <b>show pdm sessions</b> command.
<b>show pdm sessions</b>	Displays a <i>session_id</i> for each active PDM session to the PIX Firewall, beginning with session number <b>0</b> .
<b>history enable</b>	Internal PDM command. Take a data sample and store the sample data to the PDM history buffer. The <b>no</b> version of this command disables PDM data sampling.
<b>show pdm history</b>	Internal PDM command. Displays the contents of the PDM history buffer.
<b>12h</b>   <b>5d</b>   <b>60m</b>   <b>10m</b>   <b>all</b>	Specifies the PDM history view to display: 12 hours ( <b>12h</b> ), 5 days ( <b>5d</b> ), 60 minutes ( <b>60m</b> ), 10 minutes ( <b>10m</b> ), or <b>all</b> history contents in the PDM history buffer.
<b>snapshot</b>	Displays only the last PDM history data point.
<b>feature</b>	This specifies to display history for a single feature (selected with one of the following). Otherwise, all of them are displayed.
<b>blocks</b>	History for system buffers. Similar to output of the <b>show blocks</b> command.
<b>cpu</b>	History for CPU usage. Similar to output of the <b>show cpu usage</b> command.

<b>failover</b>	History for failover. Similar to output of the <b>show failover</b> command.
<b>ids</b>	History for IDS (Intrusion Detection System).
<b>memory</b>	History for memory. Similar to output of the <b>show memory</b> command.
<b>perfmon</b>	History for performance. Similar to output of <b>show perfmon</b> command.
<b>xlates</b>	History for translation slot information. Similar to output of the <b>show xlate</b> command.
<b>pdmclient</b>	Displays the PDM history in PDM-display format.
<b>location</b>	Internal PDM command. Associates an interface with an IP address on which PDM resides.
<i>ip_address</i>	Specifies the host or network on which PDM resides.
<i>netmask</i>	Specifies the network mask for the <b>pdm location ip_address</b> .
<i>if_name</i>	Specifies the interface name on which PDM resides.
<b>logging</b>	Internal PDM command. Specifies the type and number of syslog messages displayed through the PDM <b>syslog</b> option.
<i>level</i>	Specifies the priority level of syslog messages displayed in the PDM <b>syslog</b> option.
<i>messages</i>	Specifies the number of messages stored in the PDM buffer. Once the buffer is full, old messages will be discarded.
<b>show pdm logging</b>	Internal PDM command. Displays the contents of the PDM buffer within PDM.

### Defaults

Default PDM syslog *level* is **0**. Default logging *messages* is **100** and the maximum is **512**.

### Usage Guidelines

The **pdm location** command can only associate one interface to an *ip\_address/netmask* pair. Specifying an existing pair will replace the old definition. The PDM syslog messages are stored separately from the PIX Firewall syslog accessed through the **logging buffered** command.

The **clear pdm location** command will remove all of the PDM locations. The **clear pdm logging** command will clear the PDM log without disabling it.

### Examples

The following example shows how to report the last data point in PDM-display format:

```

pix(config)# show pdm history 10m snapshot pdmclient
INTERFACE|outside|up|IBC|0|OBC|1088|IPC|0|OPC|0|IBR|17|OBR|0|IPR|0|OPR|0|IERR|1|NB|0|RB|0|
RNT|0|GNT|0|CRC|0|FRM|0|OR|0|UR|0|OERR|0|COLL|0|LCOLL|0|RST|0|DEF|0|LCR|0:PIXoutsideINTERF
ACE:METRIC_HISTORY|SNAP|IBR|VIEW|10|1952|METRIC_HISTORY|SNAP|OBR|VIEW|10|64|METRIC_HISTORY
|SNAP|IPR|VIEW|10|17|METRIC_HISTORY|SNAP|OPR|VIEW|10|1|METRIC_HISTORY|SNAP|IERR|VIEW|10|0|
METRIC_HISTORY|SNAP|OERR|VIEW|10|0|:PIXinsideINTERFACE:METRIC_HISTORY|SNAP|IBR|VIEW|10|0|M
ETRIC_HISTORY|SNAP|OBR|VIEW|10|64|METRIC_HISTORY|SNAP|IPR|VIEW|10|0|METRIC_HISTORY|SNAP|OP
R|VIEW|10|1|METRIC_HISTORY|SNAP|IERR|VIEW|10|0|METRIC_HISTORY|SNAP|OERR|VIEW|10|0|:PIXSYS:
METRIC_HISTORY|SNAP|MEM|VIEW|10|52662272|METRIC_HISTORY|SNAP|BLK4|VIEW|10|1600|METRIC_HIST
ORY|SNAP|BLK80|VIEW|10|400|METRIC_HISTORY|SNAP|BLK256|VIEW|10|998|METRIC_HISTORY|SNAP|BLK1
550|VIEW|10|676|METRIC_HISTORY|SNAP|XLATES|VIEW|10|0|METRIC_HISTORY|SNAP|CONNS|VIEW|10|0|M
ETRIC_HISTORY|SNAP|TCPCONNS|VIEW|10|0|METRIC_HISTORY|SNAP|UDPCONNS|VIEW|10|0|METRIC_HISTOR
Y|SNAP|URLS|VIEW|10|0|METRIC_HISTORY|SNAP|WEBSNS|VIEW|10|0|METRIC_HISTORY|SNAP|TCPFIXUPS|V
IEW|10|0|METRIC_HISTORY|SNAP|TCPINTERCEPTS|VIEW|10|0|METRIC_HISTORY|SNAP|HTTPFIXUPS|VIEW|1
0|0|METRIC_HISTORY|SNAP|FTPFIXUPS|VIEW|10|0|METRIC_HISTORY|SNAP|AAAAUTHENUPS|VIEW|10|0|MET
RIC_HISTORY|SNAP|AAAAUTHORUPS|VIEW|10|0|METRIC_HISTORY|SNAP|AAAACCOUNTS|VIEW|10|0|

```

The following example shows how to report the last data point in non-PDM format:

```

pix(config)# show pdm history 10m snapshot
INTERFACE|outside|up|IBC|0|OBC|1344|IPC|0|OPC|0|IBR|21|OBR|0|IPR|0|OPR|0|IERR|1|NB|0|RB|0|
RNT|0|GNT|0|CRC|0|FRM|0|OR|0|UR|0|OERR|0|COLL|0|LCOLL|0|RST|0|DEF|0|LCR|0
:PIX outside INTERFACE:
Input Byte Count: [ 10s] : 1952
Output Byte Count: [ 10s] : 64
Input Packet Count: [ 10s] : 17
Output Packet Count: [ 10s] : 1
Input Error Packet Count: [ 10s] : 0
Output Error Packet Count: [ 10s] : 0
:PIX inside INTERFACE:
Input Byte Count: [ 10s] : 0
Output Byte Count: [ 10s] : 64
Input Packet Count: [ 10s] : 0
Output Packet Count: [ 10s] : 1
Input Error Packet Count: [ 10s] : 0
Output Error Packet Count: [ 10s] : 0
MEM|50479104
BLOCK|BLK4|1600|BLK80|0|BLK256|400|BLK1550|0|BLK1552|997|BLK2560|0|BLK4096|1188|BLK8192|0|
BLK16384|0|BLK65536|0
Available Memory: [ 10s] : 52662272
Available 4 bytes Blocks: [ 10s] : 1600
Available 80 bytes Blocks: [ 10s] : 400
Available 256 bytes Blocks: [ 10s] : 998
Available 1550 bytes Blocks: [ 10s] : 676
PERFMON|XLATES|0|CONNECTIONS|0|TCP CONNS|0|UDP CONNS|0|URLS|0|WEBSNS|0|TCP FIXUP|0|TCP
INTERCEPT|0|HTTP FIXUP|0|FTP FIXUP|0|AAA AUTHEN|0|AAA AUTHOR|0|AAA ACCOUNT|0
Xlate Count: [ 10s] : 0
Connection Count: [ 10s] : 0
TCP Connection Count: [ 10s] : 0
UDP Connection Count: [ 10s] : 0
URL Filtering Count: [ 10s] : 0
WEBSNSE Filtering Count: [ 10s] : 0
TCP Fixup Count: [ 10s] : 0
TCP Intercept Count: [ 10s] : 0
HTTP Fixup Count: [ 10s] : 0
FTP Fixup Count: [ 10s] : 0
AAA Authentication Count: [ 10s] : 0
AAA Authorization Count: [ 10s] : 0
AAA Accounting Count: [ 10s] : 0

```

#### Related Commands

- [copy tftp flash](#)
- [http](#)
- [setup](#)

# perfmon

View performance information. (Privileged mode.)

**perfmon interval** *seconds*

**perfmon quiet** | **verbose**

**perfmon settings**

**show perfmon**

## Syntax Description

<b>interval</b> <i>seconds</i>	Specify the number of seconds the performance display is refreshed on the console. The default is 120 seconds.
<b>quiet</b>	Disable performance monitor displays.
<b>verbose</b>	Enable displaying performance monitor information at the PIX Firewall console.
<b>settings</b>	Displays the interval and whether it is quiet or verbose.

## Usage Guidelines

The **perfmon** command allows you to monitor the PIX Firewall unit's performance. Use the **show perfmon** command to view the information immediately. Use the **perfmon verbose** command to display the information every two minutes continuously. Use the **perfmon interval** *seconds* command with the **perfmon verbose** command to display the information continuously every number of seconds you specify.



### Note

The **show perfmon** command does not display in a Telnet console session.

Use the **perfmon quiet** command to disable the display.

An example of the performance information follows:

PERFMON STATS:	Current	Average
Xlates	33/s	20/s
Connections	110/s	10/s
TCP Conns	50/s	42/s
WebSns Req	4/s	2/s
TCP Fixup	20/s	15/s
HTTP Fixup	5/s	5/s
FTP Fixup	7/s	4/s
AAA Authen	10/s	5/s
AAA Author	9/s	5/s
AAA Account	3/s	3/s

This information lists the number of translations, connections, Websense requests, address translations (called “fixups”), and AAA transactions that occur each second.

**Examples**

The following commands display the performance monitor statistics every 30 seconds on the PIX Firewall console:

```
perfmom interval 30  
perfmom verbose
```

# ping

Determine if other IP addresses are visible from the PIX Firewall. (Privileged mode.)

```
ping [if_name] ip_address
```

## Syntax Description

<i>if_name</i>	The internal or external network interface name. The address of the specified interface is used as the source address of the ping.
<i>ip_address</i>	The IP address of a host on the inside or outside networks.

## Usage Guidelines

The **ping** command determines if the PIX Firewall has connectivity or if a host is available on the network. The command output shows if the response was received; that is, that a host is participating on the network. If a host is not responding, **ping** displays “NO response received.” Use the **show interface** command to ensure that the PIX Firewall is connected to the network and is passing traffic.

If you want internal hosts to be able to ping external hosts, you must create an ICMP **access-list** command statement for echo reply; for example, to give ping access to all hosts, use the **access-list acl\_grp permit icmp any any** command and bind the **access-list** command statement to the interface you want to test using an **access-group** command statement.

If you are pinging through PIX Firewall between hosts or routers, but the pings are not successful, use the **debug icmp trace** command to monitor the success of the ping. If pings are both inbound and outbound, they are successful.

The PIX Firewall **ping** command no longer requires an interface name. If an interface name is not specified, PIX Firewall checks the routing table to find the address you specify. You can specify an interface name to indicate through which interface the ICMP echo requests are sent.

An example of the usage follows:

```
ping 10.0.0.1
10.0.0.1 response received -- 10ms
10.0.0.1 response received -- 10ms
10.0.0.1 response received -- 0ms
```

Or you can still enter the command specifying the interface:

```
ping outside 10.0.0.1
10.0.0.1 response received -- 10ms
10.0.0.1 response received -- 10ms
10.0.0.1 response received -- 0ms
```

## Examples

The **ping** command makes three attempts to reach an IP address:

```
ping 192.168.42.54
192.168.42.54 response received -- 0ms
192.168.42.54 response received -- 0ms
192.168.42.54 response received -- 0ms
```

# quit

Exit configuration or privileged mode. (All modes.)

## quit

---

### Usage Guidelines

Use the **quit** command to exit configuration or privileged mode.

---

### Examples

The following example shows use of the **quit** command:

```
pixfirewall(config)# quit
pixfirewall# quit
pixfirewall>
```

# reload

Reboot and reload the configuration. (Privileged mode.)

**reload**

**reload noconfirm**

## Syntax Description

<b>reload</b>	Reboot and reload configuration.
<b>noconfirm</b>	Permits the PIX Firewall to reload without user confirmation.

## Usage Guidelines

The **reload** command reboots the PIX Firewall and reloads the configuration from a bootable floppy disk or, if a diskette is not present, from Flash memory.

The PIX Firewall does not accept abbreviations to the keyword **noconfirm**.

You are prompted for confirmation before starting with “Proceed with reload?”.

Any response other than **n** causes the reboot to occur.



### Note

Configuration changes not written to Flash memory are lost after reload. Before rebooting, store the current configuration in Flash memory with the **write memory** command.

## Examples

The following example shows use of the **reload** command:

```

reload
Proceed with reload? [confirm] y

Rebooting...

PIX Bios V2.7
...

```

# rip

Change RIP settings. (Configuration mode.)

```
rip if_name default | passive [version [1 | 2]] [authentication [text | md5 key (key_id)]]
```

```
no rip if_name default | passive [version [1 | 2]] [authentication [text | md5 key (key_id)]]
```

```
show rip [if_name]
```

```
clear rip
```

```
debug rip [if_name]
```

## Syntax Description

<i>if_name</i>	The internal or external network interface name.
<b>default</b>	Broadcast a default route on the interface.
<b>passive</b>	Enable passive RIP on the interface. The PIX Firewall listens for RIP routing broadcasts and uses that information to populate its routing tables.
<b>version</b>	RIP version. Use <b>version 2</b> for RIP update encryption. Use <b>version 1</b> to provide backward compatibility with the older version.
<b>authentication</b>	Enable RIP version 2 authentication.
<b>text</b>	Send RIP updates as clear text (not recommended).
<b>md5</b>	Send RIP updates using MD5 encryption.
<i>key</i>	Key to encrypt RIP updates. This value must be the same on the routers and any other device <i>that provides RIP version 2 updates</i> . The <i>key</i> is a text string of up to 16 characters in length.
<i>key_id</i>	Key identification value. The <i>key_id</i> can be a number from 1 to 255. Use the same <i>key_id</i> in use on the routers and any other device <i>that provides RIP version 2 updates</i> .

## Usage Guidelines

The **rip** command enables IP routing table updates from received Routing Information Protocol (RIP) broadcasts. Use the **show rip** command to display the current RIP settings. Use the **no rip** command to disable the PIX Firewall IP routing table updates. The default is to enable IP routing table updates. If you specify RIP version 2, you can encrypt RIP updates using MD5 encryption.

The **clear rip** command removes all the **rip** commands from the configuration.

Ensure that the *key* and *key\_id* values are the same as in use on any other device in your network that makes RIP version 2 updates.

The PIX Firewall cannot pass RIP updates between interfaces.

When RIP version 2 is configured in passive mode with PIX Firewall software version 5.3 and higher, the PIX Firewall accepts RIP version 2 multicast updates with IP destination of 224.0.0.9. For RIP version 2 default mode, the PIX Firewall will transmit default route updates using an IP destination of 224.0.0.9. Configuring RIP version 2 registers the multicast address 224.0.0.9 on the respective interface to be able to accept multicast RIP version 2 updates.

Only Intel 10/100 and Gigabit interfaces support multicasting. FDDI and Token Ring will still operate in broadcast mode (IP destination 255.255.255.255 not 224.0.0.9).

When the RIP version 2 commands for an interface are removed, the multicast address is unregistered from the interface card.

### Examples

The following is sample output from the version 1 **show rip** and **rip inside default** commands:

```
show rip
rip outside passive
no rip outside default
rip inside passive
no rip inside default
```

```
rip inside default
show rip
rip outside passive
no rip outside default
rip inside passive
rip inside default
```

The next example combines version 1 and version 2 commands and shows listing the information with the **show rip** command after entering the rip commands that:

- Enable version 2 passive RIP using MD5 authentication on the outside interface to encrypt the key used by the PIX Firewall and other RIP peers, such as routers.
- Enable version 1 passive RIP listening on the inside interface of the PIX Firewall.
- Enable version 2 passive RIP listening on the dmz interface of the PIX Firewall.

```
rip outside passive version 2 authentication md5 thisisakey 2
rip outside default version 2 authentication md5 thisisakey 2
rip inside passive
rip dmz passive version 2

show rip
rip outside passive version 2 authentication md5 thisisakey 2
rip outside default version 2 authentication md5 thisisakey 2
rip inside passive version 1
rip dmz passive version 2
```

The next example shows how use of the **clear rip** command clears all the previous **rip** commands from the current configuration:

```
clear rip
show rip
```

This example shows use of the version 2 feature that passes the encryption key in text form:

```
rip out default version 2 authentication text thisisakey 3
show rip
rip outside default version 2 authentication text thisisakey 3
```

# route

Enter a static or default route for the specified interface. (Configuration mode.)

**route** *if\_name ip\_address netmask gateway\_ip [metric]*

**clear route** [*if\_name ip\_address [netmask gateway\_ip]*]

**no route** [*if\_name ip\_address [netmask gateway\_ip]*]

**show route**

## Syntax Description

<i>if_name</i>	The internal or external network interface name.
<i>ip_address</i>	The internal or external network IP address. Use <b>0.0.0.0</b> to specify a default route. The <b>0.0.0.0</b> IP address can be abbreviated as <b>0</b> .
<i>netmask</i>	Specify a network mask to apply to <i>ip_address</i> . Use <b>0.0.0.0</b> to specify a default route. The <b>0.0.0.0</b> <i>netmask</i> can be abbreviated as <b>0</b> .
<i>gateway_ip</i>	Specify the IP address of the gateway router (the next hop address for this route).
<i>metric</i>	Specify the number of hops to <i>gateway_ip</i> . If you are not sure, enter <b>1</b> . Your network administrator can supply this information or you can use a <b>tracert</b> command to obtain the number of hops. The default is <b>1</b> if a metric is not specified.

## Usage Guidelines

Use the **route** command to enter a default or static route for an interface. To enter a default route, set *ip\_address* and *netmask* to **0.0.0.0**, or the shortened form of **0**. All routes entered using the **route** command are stored in the configuration when it is saved. The **clear route** command removes **route** command statements from the configuration that do not contain the CONNECT keyword.

Create static routes to access networks connected outside a router on any interface. The effect of a static route is like stating “to send a packet to the specified network, give it to this router.” For example, PIX Firewall sends all packets destined to the 192.168.42.0 network through the 192.168.1.5 router with this static **route** command statement.

```
route dmz 192.168.42.0 255.255.255.0 192.168.1.5 1
```

The routing table automatically specifies the IP address of a PIX Firewall interface in the **route** command. Once you enter the IP address for each interface, PIX Firewall creates a **route** statement entry that is not deleted when you use the **clear route** command.

If the **route** command statement uses the IP address from one of the PIX Firewall unit’s interfaces as the gateway IP address, PIX Firewall will ARP for the destination IP address in the packet instead of ARPing for the gateway IP address.

The following steps show how PIX Firewall handles routing:

- 
- Step 1** PIX Firewall receives a packet from the inside interface destined to IP address X.
  - Step 2** Because a default route is set to itself, PIX Firewall sends out an ARP for address X.
  - Step 3** Any Cisco router on the outside interface LAN which has a route to address X (Cisco IOS software has proxy ARP enabled by default) replies back to the PIX Firewall with its own MAC address as the next hop.
  - Step 4** PIX Firewall sends the packet to router (just like a default gateway).

- Step 5** PIX Firewall adds the entry to its ARP cache for IP address X with the MAC address being that of the router.
- The CONNECT route entry is supported. (This identifier appears when you use the **show route** command.) The CONNECT identifier is assigned to an interface's local network and the interface IP address, which is in the IP local subnet. PIX Firewall will ARP for the destination address. The CONNECT identifier cannot be removed, but changes when you change the IP address on the interface.
  - If you enter duplicate routes with different metrics for the same gateway, PIX Firewall changes the metric for that route and updates the metric for the route.

For example, the following command statement is in a configuration:

```
route inside 10.0.0.0 255.0.0.0 10.0.0.2 2 OTHER
```

If you enter the following statement:

```
route inside 10.0.0.0 255.0.0.0 10.0.0.2 3
```

PIX Firewall converts the command statement to the following:

```
route inside 10.0.0.0 255.0.0.0 10.0.0.2 3 OTHER
```

---

## Examples

Specify one default **route** command statement for the outside interface, which in this example, is for the router on the outside interface that has an IP address of 209.165.201.1:

```
route outside 0 0 209.165.201.1 1
```

For static routes, if two networks, 10.1.2.0 and 10.1.3.0 connect via a hub to the dmz1 interface router at 10.1.1.4, add these static **route** command statements to provide access to the networks:

```
route dmz1 10.1.2.0 255.0.0.0 10.1.1.4 1
route dmz1 10.1.3.0 255.0.0.0 10.1.1.4 1
```