



## D through F Commands

---

### debug

You can debug packets or ICMP tracings through the PIX Firewall. The **debug** command provides information that helps troubleshoot protocols operating with and through the PIX Firewall.

[no] debug aaa [authentication | authorization | accounting | internal]

[no] debug access-list all | standard | turbo

[no] debug arp

[no] debug crypto ca [level]

[no] debug ctique

[no] debug crypto ipsec [level]

[no] debug crypto isakmp [level]

[no] debug crypto vpnclient

[no] debug dhcp detail | error | packet

[no] debug dhcpd event | packet

[no] debug dhcprelay event | packet | error

[no] debug dns {resolver | all}

[no] debug fixup {udp | tcp}

[no] debug fover *option*

[no] debug h323 h225 [asn | event]

[no] debug h323 h245 [asn | event]

[no] debug h323 ras [asn | event]

[no] debug icmp trace

[no] debug ils

```

[no] debug ospf [adj | database-timer | events | flood | lsa-generation | packet | tree |
retransmission | spf [external | internal | intra]]

[no] debug mgcp [messages | parser | sessions]

[no] debug ntp [adjust | authentication | events | loopfilter | packets | params | select | sync |
validity]

[no] debug packet if_name [src source_ip [netmask mask]] [dst dest_ip [netmask mask]] [[proto
icmp] | [proto tcp [sport src_port] [dport dest_port]] | [proto udp [sport src_port] [dport
dest_port]]] [rx | tx | both]

[no] debug pdm history

[no] debug ppp error | io | uauth | upap | chap | negotiation

[no] debug pppoe event | error | packet

[no] debug pptp

[no] debug radius [session | all | user username]

[no] debug rip

[no] debug route

[no] debug rtsp

[no] debug sip

[no] debug skinny

[no] debug sqlnet

[no] debug ssh

[no] debug ssl [cypher | device]

[no] debug vpdn event | error | packet

[no] debug xdmcp

no debug all

undebug all

show debug

```

**Syntax Description**

<b>aaa</b>	Displays authentication, authorization, and accounting information.
<b>access-list</b>	Displays access list configuration information.
<b>adjust</b>	Displays NTP clock adjustments.
<b>all</b>	Displays both standard and TurboACL access list information.
<b>authentication</b>	Displays NTP clock authentication.

<b>both</b>	Displays both received and transmitted packets.
<b>chap</b>	Displays CHAP/MS-CHAP authentication.
<b>crypto ca</b>	Displays information about certification authority (CA) traffic.
<b>crypto ipsec</b>	Displays information about IPSec traffic.
<b>crypto isakmp</b>	Displays information about IKE traffic.
<b>crypto vpnclient</b>	Displays information about the firewall EasyVPN client.
<b>ctiqbe</b>	Displays information about CTI Quick Buffer Encoding (CTIQBE), which is used with Cisco TAPI/JTAPI applications.
<b>cypher</b>	Display information about the cipher negotiation between the HTTP server and the client.
<b>device</b>	Displays information about the SSL device including session initiation and ongoing status.
<b>dhcpc detail</b>	Displays detailed information about the DHCP client packets.
<b>dhcpc error</b>	Displays error messages associated with the DHCP client.
<b>dhcpc packet</b>	Displays packet information associated with the DHCP client.
<b>dhcpcd event</b>	Displays event information associated with the DHCP server.
<b>dhcpcd packet</b>	Displays packet information associated with the DHCP server.
<b>dhcprelay</b>	Displays DHCP Relay Agent information.
<b>dns {resolver   all}</b>	Displays DNS debugging information. The <b>resolver</b> option collects DNS resolution information, and the <b>all</b> option collects all DNS information.
<b>dport</b> <i>dest_port</i>	Destination port.
<b>dst</b> <i>dest_ip</i>	Destination IP address.
<b>events</b>	Displays NTP event information.
<b>fixup {udp   tcp}</b>	Displays fixup information, using either UDP or TCP.
<b>fover</b> <i>option</i>	Displays failover information. Refer to <a href="#">Table 5-1</a> for the <i>options</i> .
<b>h225 asn</b>	Displays the output of the decoded PDUs.
<b>h225 events</b>	Displays the events of the H.225 signaling, or turn both traces on.
<b>h245 asn</b>	Displays the output of the decoded PDUs.
<b>h245 events</b>	Displays the events of the H.245 signaling, or turn both traces on.
<b>h323</b>	Displays information about the packet-based multimedia communications systems standard.
<b>icmp</b>	Displays information about ICMP traffic.
<b>if_name</b>	Interface name from which the packets are arriving; for example, to monitor packets coming into the PIX Firewall from the outside, set <i>if_name</i> to <b>outside</b> .
<b>ils</b>	Displays Internet Locator Service (ILS) fixup information (used in LDAP services).

<i>level</i>	<p>The level of debugging feedback. The higher the level number, the more information is displayed. The default <i>level</i> is 1. The levels correspond to the following events:</p> <ul style="list-style-type: none"> <li>• Level 1: Interesting events</li> <li>• Level 2: Normative and interesting events</li> <li>• Level 3: Diminutive, normative, and interesting events</li> </ul> <p>Refer to the “Examples” section at the end of this command page for an example of how the debugging level appears within the <b>show debug</b> command.</p>
loopfilter	Displays NTP loop filter information.
<b>messages</b>	Displays debug information for MGCP messages.
negotiation	Equivalent of the <b>error</b> , <b>uauth</b> , <b>upap</b> and <b>chap debug</b> command options.
<b>netmask</b> <i>mask</i>	Network mask.
<b>packet</b>	Displays packet information.
packets	Displays NTP packet information.
params	Displays NTP clock parameters.
<b>parser</b>	Displays debug information about parsing MGCP messages.
<b>pdm history</b>	Turns on the PDM history metrics debugging information. The <b>no</b> version of this command disables PDM history metrics debugging.
<b>ppp</b>	Debugs L2TP or PPTP traffic, which is configured with the <b>vpdn</b> command.
<b>ppp error</b>	Displays L2TP or PPTP PPP virtual interface error messages.
<b>ppp io</b>	Display the packet information for L2TP or PPTP PPP virtual interface.
<b>ppp uauth</b>	Displays the L2TP or PPTP PPP virtual interface AAA user authentication debugging messages.
pppoe error	Displays PPPoE error messages.
pppoe event	Displays PPPoE event information.
pppoe packet	Displays PPPoE packet information.
pptp	Displays PPTP traffic information.
<b>proto icmp</b>	Displays ICMP packets only.
<b>proto tcp</b>	Displays TCP packets only.
<b>proto udp</b>	Displays UDP packets only.
radius all	Enables all RADIUS debug options.
radius session	Logs RADIUS session information and the attributes of sent and received RADIUS packets.
<b>ras asn</b>	Displays the output of the decoded PDUs.
<b>ras events</b>	Displays the events of the RAS signaling, or turn both traces on.
route	Displays information from the PIX Firewall routing module.
<b>rx</b>	Displays only packets received at the PIX Firewall.
select	Displays NTP clock selections.
<b>sessions</b>	Displays debug information for MGCP sessions.
sip	Debug the fixup Session Initiation Protocol (SIP) module.
<b>skinny</b>	Debugs SCCP protocol activity. (Using this option is system-resources intensive and may impact performance on high traffic network segments.)

<b>sport</b> <i>src_port</i>	Source port. See the “Ports” section in “Chapter 2, “Using PIX Firewall Commands” for a list of valid port literal names.
<b>sqlnet</b>	Debugs SQL*Net traffic.
<b>src</b> <i>source_ip</i>	Source IP address.
<b>ssh</b>	Debug information and error messages associated with the <b>ssh</b> command.
<b>ssl</b>	Debug information and error messages associated with the <b>ssl</b> command.
<b>standard</b>	Displays non-TurboACL access list information.
<b>sync</b>	Displays NTP clock synchronization.
<b>turbo</b>	Displays TurboACL access list information.
<b>tx</b>	Displays only packets that were transmitted from the PIX Firewall.
<b>upap</b>	Displays PAP authentication.
<b>user</b> <i>username</i>	Specifies to display information for an individual <i>username</i> only.
<b>validity</b>	Displays NTP peer clock validity.
<b>vpdn error</b>	Display L2TP or PPTP protocol error messages.
<b>vpdn event</b>	Display L2TP or PPTP tunnel event change information.
<b>vpdn packet</b>	Display L2TP or PPTP packet information about PPTP traffic.
<b>xmcp</b>	Display information about the xmcp negotiation

### Defaults

MGCP debugging is disabled by default.

### Command Modes

Configuration mode unless otherwise specified.

The **debug mgcp** command is available in privileged mode.

### Usage Guidelines

The **debug** command lets you view debug information. The **show debug** command displays the current state of tracing. You can debug the contents of network layer protocol packets with the **debug packet** command.



#### Note

Use of the **debug** commands may slow down traffic on busy networks.

Use of the **debug packet** command on a PIX Firewall experiencing a heavy load may result in the output displaying so fast that it may be impossible to stop the output by entering the **no debug packet** command from the console. You can enter the **no debug packet** command from a Telnet session.

To let users ping through the PIX Firewall, add the **access-list acl\_grp permit icmp any any** command statement to the configuration and bind it to each interface you want to test with the **access-group** command. This lets pings go outbound and inbound.

To stop a **debug packet trace** command, enter the following command:

```
no debug packet if_name
```

Replace *if\_name* with the name of the interface; for example, **inside**, **outside**, or a perimeter interface name.

**no debug all and undebug all**

The **no debug all** and **undebug all** commands stop any and all debug messages from being displayed.

**debug crypto**

When creating your digital certificates, use the **debug crypto ca** command to ensure that the certificate is created correctly. Important error messages only display when the **debug crypto ca** command is enabled. For example, if you enter an Entrust fingerprint value incorrectly, the only warning message that indicates the value is incorrect appears in the **debug crypto ca** command output.

Output from the **debug crypto ipsec** and **debug crypto isakmp** commands does not display in a Telnet console session.

**debug dhcpc**

The **debug dhcpc detail** command displays detailed packet information about the DHCP client. The **debug dhcpc error** command displays DHCP client error messages. The **debug dhcpc packet** command displays packet information about the DHCP client. Use the **no** form of the **debug dhcpc** command to disable debugging.

The **debug dhcpd event** command displays event information about the DHCP server. The **debug dhcpd packet** command displays packet information about the DHCP server. Use the **no** form of the **debug dhcpd** commands to disable debugging.

**debug h323**

The **debug h323** command lets you debug H.323 connections. Use the **no** form of the command to disable debugging. This command works when the **fixup protocol h323** command is enabled.

**Note**

The **debug h323** command, particularly the **debug h323 h225 asn**, **debug h323 h245 asn**, and **debug h323 ras asn** commands, might delay the sending of messages and cause slower performance in a real-time environment.

**debug icmp**

The **debug icmp trace** command shows ICMP packet information, the source IP address, and the destination address of packets arriving, departing, and traversing the PIX Firewall including pings to the PIX Firewall unit's own interfaces.

To stop a **debug icmp trace** command, enter the following command:

```
no debug icmp trace
```

**debug mgcp**

The **debug mgcp** command displays debug information for Media Gateway Control Protocol (MGCP) traffic. Without any options explicitly specified, the **debug mgcp** command enables all three MGCP debug options. The **no debug mgcp** command, without any options explicitly specified, disables all MGCP debugging.

**debug ospf**

The **debug ospf** command enables all OSPF debugging options, and the **no debug ospf** command disables all OSPF debugging options.

The **debug ospf spf** command enables all SPF options, and the **no debug ospf spf** command disables all SPF options.

**debug sqlnet**

The **debug sqlnet** command reports on traffic between Oracle SQL\*Net clients and servers through the PIX Firewall.

**debug ssh**

The **debug ssh** command reports on information and error messages associated with the **ssh** command.

**debug pptp**

The **debug pptp** and **debug vpdn** commands provide information about PPTP traffic. PPTP is configured with the **vpdn** command.

**debug fover**

Table 5-1 lists the options for the **debug fover** command.

**Table 5-1** *debug fover Command Options*

Option	Description
cable	Failover cable status
fail	Failover internal exception
fmsg	Failover message
get	IP network packet received
ifc	Network interface status trace
lanrx	LAN-based failover receive process messages
lanretx	LAN-based failover retransmit process messages
lantx	LAN-based failover transmit process messages
lancmd	LAN-based failover main thread messages
open	Failover device open
put	IP network packet transmitted
rx	Failover cable receive
rxdump	Cable rcv message dump (serial console only)
rxip	IP network failover packet received
tx	Failover cable transmit
txdump	Cable xmit message dump (serial console only)
txip	IP network failover packet transmit
verify	Failover message verify
switch	Failover Switching status

**Trace Channel Feature**

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

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

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

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

The **debug** commands, except the debug crypto commands, are shared between all Telnet and serial console sessions.



#### Note

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

#### Examples

The following is partial sample output from the **debug dhcpc packet** and the **debug dhcpc detail** commands. The **ip address dhcp setroute** command was configured after entering the **debug dhcpc** commands to obtain debugging information.

```
debug dhcpc packet
debug dhcpc detail
ip address outside dhcp setroute
DHCP:allocate request
DHCP:new entry. add to queue
DHCP:new ip lease str = 0x80ce8a28
DHCP:SDiscover attempt # 1 for entry:
Temp IP addr:0.0.0.0 for peer on Interface:outside
Temp sub net mask:0.0.0.0
    DHCP Lease server:0.0.0.0, state:1 Selecting
    DHCP transaction id:0x8931
    Lease:0 secs, Renewal:0 secs, Rebind:0 secs
    Next timer fires after:2 seconds
    Retry count:1   Client-ID:cisco-0000.0000.0000-outside

DHCP:SDiscover:sending 265 byte length DHCP packet
DHCP:SDiscover 265 bytes
DHCP Broadcast to 255.255.255.255 from 0.0.0.0
DHCP client msg received, fip=10.3.2.2, fport=67
DHCP:Received a BOOTREP pkt
DHCP:Scan:Message type:DHCP Offer
DHCP:Scan:Server ID Option:10.1.1.69 = 450A44AB
    DHCP:Scan:Server ID Option:10.1.1.69 = 450A44AB
DHCP:Scan:Lease Time:259200
DHCP:Scan:Subnet Address Option:255.255.254.0
DHCP:Scan:DNS Name Server Option:10.1.1.70, 10.1.1.140
DHCP:Scan:Domain Name:example.com
```



```
DHCP:Scan:NBNS Name Server Option:10.1.2.228, 10.1.2.87
DHCP:Scan:Router Address Option:10.3.2.1
DHCP:rcvd pkt source:10.3.2.2, destination: 255.255.255.255
...
```

The following example executes the **debug icmp trace** command:

```
debug icmp trace
```

When you ping a host through the PIX Firewall from any interface, trace output displays on the console. The following example shows a successful ping from an external host (209.165.201.2) to the PIX Firewall unit's outside interface (209.165.201.1).

```
Inbound ICMP echo reply (len 32 id 1 seq 256) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 512) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 512) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 768) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 768) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 1024) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 1024) 209.165.201.1 > 209.165.201.2
NO DEBUG ICMP TRACE
ICMP trace off
```

This example shows that the ICMP packet length is 32 bytes, the ICMP packet identifier is 1, and the ICMP sequence number. The ICMP sequence number starts at 0 and is incremented each time a request is sent.

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

```
show debug
debug ppp error
debug vpdn event
debug crypto ipsec 1
debug crypto isakmp 1
debug crypto ca 1
debug icmp trace
debug packet outside both
debug sqlnet
```

The preceding sample output includes the **debug crypto** commands.

The following example shows debugging messages for Unity client negotiation using Diffie-Hellman group 5:

```
pixfirewall(config)# debug crypto isakmp

check_isakmp_proposal:
is_auth_policy_configured: auth 1
is_auth_policy_configured: auth 4
ISAKMP (0): Checking ISAKMP transform 1 against priority 8 policy
ISAKMP:      encryption 3DES-CBC
ISAKMP:      hash SHA
ISAKMP:      default group 5
ISAKMP:      extended auth RSA sig
ISAKMP:      life type in seconds
ISAKMP:      life duration (VPI) of  0x0 0x20 0xc4 0x9b
ISAKMP (0): atts are not acceptable. Next payload is 3
ISAKMP (0): Checking ISAKMP transform 2 against priority 8 policy
ISAKMP:      encryption 3DES-CBC
ISAKMP:      hash MD5
ISAKMP:      default group 5
ISAKMP:      extended auth RSA sig
ISAKMP:      life type in seconds
ISAKMP:      life duration (VPI) of  0x0 0x20 0xc4 0x9b
```

```

ISAKMP (0): atts are not acceptable. Next payload is 3
ISAKMP (0): Checking ISAKMP transform 3 against priority 8 policy
ISAKMP:      encryption 3DES-CBC
ISAKMP:      hash SHA
ISAKMP:      default group 5
ISAKMP:      auth RSA sig
ISAKMP:      life type in seconds
ISAKMP:      life duration (VPI) of  0x0 0x20 0xc4 0x9b
ISAKMP (0): atts are not acceptable. Next payload is 3
ISAKMP (0): Checking ISAKMP transform 4 against priority 8 policy
ISAKMP:      encryption 3DES-CBC
ISAKMP:      hash MD5
ISAKMP:      default group 5
ISAKMP:      auth RSA sig
ISAKMP:      life type in seconds
ISAKMP:      life duration (VPI) of  0x0 0x20 0xc4 0x9b
ISAKMP (0): atts are acceptable. Next payload is 3

```

The following example shows possible output for the **debug mgcp messages** command:

```

17: MGCP: Retransmitted command RSIP
      Gateway IP      gate-1
      Transaction ID  1
18: MGCP: Expired command RSIP
      Gateway IP      gate-1
      Transaction ID  1
19: MGCP: New command RSIP
      Gateway IP      gate-1
      Transaction ID  1
      Endpoint name   d001
      Call ID
      Connection ID
      Media IP        0.0.0.0
      Media port      0
      Flags            0x80
20: MGCP: Retransmitted command RSIP
      Gateway IP      gate-1
      Transaction ID  1

```

The following example shows possible output for the **debug mgcp parser** command:

```

28: MGCP packet:
RSIP 1 d001@10.10.10.11 MGCP 1.0
RM: restart

29: MGCP: command verb - RSIP
30: MGCP: transaction ID - 1
31: MGCP: endpoint name - d001
32: MGCP: header parsing succeeded
33: MGCP: restart method - restart
34: MGCP: payload parsing succeeded
35: MGCP packet:
RSIP 1 d001@10.10.10.11 MGCP 1.0
RM: restart

36: MGCP: command verb - RSIP
37: MGCP: transaction ID - 1
38: MGCP: endpoint name - d001
39: MGCP: header parsing succeeded
40: MGCP: restart method - restart
41: MGCP: payload parsing succeeded

```

The following example shows possible output for the **debug mgcp sessions** command:

```

91: NAT::requesting UDP conn for generic-pc-2/6166 [209.165.202.128/0]
    from dmz/ca:generic-pc-2/2427 to outside:generic-pc-1/2727
92: NAT::reverse route: embedded host at dmz/ca:generic-pc-2/6166
93: NAT::table route: embedded host at outside:209.165.202.128/0
94: NAT::pre-allocate connection for outside:209.165.202.128 to dmz/ca:generic-pc-2/6166
95: NAT::found inside xlate from dmz/ca:generic-pc-2/0 to outside:209.165.201.15/0
96: NAT::outside NAT not needed
97: NAT::created UDP conn dmz/ca:generic-pc-2/6166 <-> outside:209.165.202.128/0
98: NAT::created RTCP conn dmz/ca:generic-pc-2/6167 <-> outside:209.165.202.128/0
99: NAT::requesting UDP conn for 209.165.202.128/6058 [generic-pc-2/0]
    from dmz/ca:generic-pc-2/2427 to outside:generic-pc-1/2727
100: NAT::table route: embedded host at outside:209.165.202.128/6058
101: NAT::reverse route: embedded host at dmz/ca:generic-pc-2/0
102: NAT::pre-allocate connection for dmz/ca:generic-pc-2 to outside:209.165.202.128/6058
103: NAT::found inside xlate from dmz/ca:generic-pc-2/0 to outside:209.165.201.15/0
104: NAT::outside NAT not needed
105: NAT::created UDP conn dmz/ca:generic-pc-2/0 <-> outside:209.165.202.128/6058
106: NAT::created RTCP conn dmz/ca:generic-pc-2/0 <-> outside:209.165.202.128/6058
107: MGCP: New session
    Gateway IP      generic-pc-2
    Call ID         9876543210abcdef
    Connection ID    6789af54c9
    Endpoint name    aaln/1
    Media lcl port   6166
    Media rmt IP     209.165.202.128
    Media rmt port    6058
108: MGCP: Expired session, active 0:06:05
    Gateway IP      generic-pc-2
    Call ID         9876543210abcdef
    Connection ID    6789af54c9
    Endpoint name    aaln/1
    Media lcl port   6166
    Media rmt IP     209.165.202.128
    Media rmt port    6058

```

You can debug the contents of packets with the **debug packet** command:

#### debug packet inside

```

----- PACKET -----
-- IP --
4.3.2.1 ==>      255.3.2.1
    ver = 0x4      hlen = 0x5      tos = 0x0      tlen = 0x60
    id = 0x3902     flags = 0x0     frag off=0x0
    ttl = 0x20      proto=0x11     chksum = 0x5885
-- UDP --
    source port = 0x89      dest port = 0x89
    len = 0x4c      checksum = 0xa6a0
-- DATA --
    00000014:                                00 01 00 00 |
    ....
    00000024: 00 00 00 01 20 45 49 45 50 45 47 45 47 45 46 46 | ..
.. EIEPEGEGEFF
    00000034: 43 43 4e 46 41 45 44 43 41 43 41 43 41 43 41 43 | CC
NFAEDCACACACAC
    00000044: 41 43 41 41 41 00 00 20 00 01 c0 0c 00 20 00 01 | AC
AAA.. .....
    00000054: 00 04 93 e0 00 06 60 00 01 02 03 04 00          | ..
....\.....
----- END OF PACKET -----

```

This display lists the information as it appears in a packet.

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

```
show debug
debug icmp trace off
debug packet off
debug sqlnet off
```

#### Related Commands

<a href="#">mgcp</a>	Configures additional support for the Media Gateway Control Protocol fixup (packet application inspection) and is used with the <b>fixup protocol mgcp</b> command.
<a href="#">show conn</a>	Displays all active connections. There is an MGCP <b>show conn</b> option and connection flag, “g”.
<a href="#">timeout</a>	Sets the maximum idle time duration. (There is an MGCP timeout option.)

## dhcpd

Configures the DHCP server.

```
[no] dhcpd address ip1[-ip2] if_name
[no] dhcpd auto_config [outside]
[no] dhcpd dns dns1 [dns2]
[no] dhcpd wins wins1 [wins2]
[no] dhcpd lease lease_length
[no] dhcpd domain domain_name
[no] dhcpd enable if_name
[no] dhcpd option 66 ascii {server_name | server_ip_str}
[no] dhcpd option 150 ip server_ip1 [ server_ip2]
no dhcpd option code
[no] dhcpd ping_timeout timeout
[no] debug dhcpd event
[no] debug dhcpd packet
clear dhcpd [binding|statistics]
show dhcpd [binding|statistics]
```

Syntax	Description
<b>address</b> <i>ip1</i> [ <i>ip2</i> ]	The IP pool address range. The size of the pool is limited to 32 addresses with a 10-user license and 128 addresses with a 50-user license on the PIX 501. The unlimited user license on the PIX 501 and all other PIX Firewall platforms support 256 addresses.  If the address pool range is larger than 253 addresses, the netmask of the PIX Firewall interface cannot be a Class C address (for example, 255.255.255.0) and hence needs to be something larger, for example, 255.255.254.0.
<b>auto_config</b>	Enable PIX Firewall to automatically configure DNS, WINS and domain name values from the DHCP client to the DHCP server. If the user also specifies <b>dns</b> , <b>wins</b> , and <b>domain</b> parameters, then the CLI parameters overwrite the auto_config parameters.
<b>binding</b>	The binding information for a given server IP address and its associated client hardware address and lease length.
<i>code</i>	Specifies the DHCP option code, either 66 or 150.
<b>dns</b> <i>dns1</i> [ <i>dns2</i> ]	The IP addresses of the DNS servers for the DHCP client. Specifies that DNS A (address) resource records that match the static translation are rewritten. A second server address is optional.
<b>domain</b> <i>domain_name</i>	The DNS domain name. For example, <b>example.com</b> .
<i>if_name</i>	Specifies the interface on which to enable the DHCP server.
<b>lease</b> <i>lease_length</i>	The length of the lease, in seconds, granted to DHCP client from the DHCP server. The lease indicates how long the client can use the assigned IP address. The default is 3600 seconds. The minimum lease length is 300 seconds, and the maximum lease length is 2,147,483,647 seconds.
<b>option 150</b>	Specifies the TFTP server IP address(es) designated for Cisco IP Phones in dotted-decimal format. DHCP option 150 is site-specific; it gives the IP addresses of a list of TFTP servers.
<b>option 66</b>	Specifies the TFTP server IP address designated for Cisco IP Phones and gives the IP address or the host name of a single TFTP server.
<b>outside</b>	The <b>outside</b> interface of the firewall.
<i>ping_timeout</i>	Allows the configuration of the timeout value of a ping, in milliseconds, before assigning an IP address to a DHCP client.
<i>server_ip(1,2)</i>	Specifies the IP address(es) of a TFTP server.
<i>server_ip_str</i>	Specifies the TFTP server in dotted-decimal format, such as 1.1.1.1, but is treated as a character string by the PIX Firewall DHCP server.
<i>server_name</i>	Specifies an ASCII character string representing the TFTP server.
<b>statistics</b>	Statistical information, such as address pool, number of bindings, malformed messages, sent messages, and received messages.
<b>wins</b> <i>wins1</i> [ <i>wins2</i> ]	The IP addresses of the Microsoft NetBIOS name servers (WINS server). The second server address is optional.

**Command Modes** Configuration mode.

## Usage Guidelines

A DHCP server provides network configuration parameters to a DHCP client. Support for the DHCP server within the PIX Firewall means the PIX Firewall can use DHCP to configure connected clients. This DHCP feature is designed for the remote home or branch office that will establish a connection to an enterprise or corporate network. See the *Cisco PIX Firewall and VPN Configuration Guide* for information on how to implement the DHCP server feature into the PIX Firewall.

You must specify an interface name, *if\_name*, for all DHCP server commands when using PIX Firewall software Version 6.3. In earlier software versions, only the inside interface could be configured as the DHCP server so there was no need to specify *if\_name*.



### Note

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

The **dhcpcd address** *ip1*[-*ip2*] *if\_name* command specifies the DHCP server address pool. The address pool of a PIX Firewall DHCP server must be within the same subnet of the PIX Firewall interface that is enabled and you must specify the associated PIX Firewall interface with the *if\_name*. In other words, the client must be physically connected to the subnet of a PIX Firewall interface. The size of the pool is limited to 32 addresses with a 10-user license and 128 addresses with a 50-user license on the PIX 501. The unlimited user license on the PIX 501 and all other PIX Firewall platforms support 256 addresses.



### Note

When the PIX Firewall responds to a DHCP client request, it uses the IP address of the interface where the request was received as the default gateway in the response. It uses the subnet mask on that interface for the subnet mask in its response.

Use caution with names that contain a “-” (dash) character because the **dhcpcd address** command interprets the last (or only) “-” character in the name as a range specifier instead of as part of the name. For example, the **dhcpcd address** 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 **no dhcpcd address** command removes the DHCP server address pool you configured.

The **dhcpcd dns** command specifies the IP address(es) of the DNS server(s) for DHCP client. You have the option to specify two DNS servers. The **no dhcpcd dns** command removes the DNS IP address(es) from your configuration.

The **dhcpcd wins** command specifies the addresses of the WINS server for the DHCP client. The **no dhcpcd dns** command removes the WINS server IP address(es) from your configuration.

The **dhcpcd lease** command specifies the length of the lease in seconds granted to the DHCP client. This lease indicates how long the DHCP client can use the assigned IP address the DHCP granted. The **no dhcpcd lease** command removes the lease length that you specified from your configuration and replaces this value with the default value of 3600 seconds.

The **dhcpcd domain** command specifies the DNS domain name for the DHCP client. For example, **example.com**. The **no dhcpcd domain** command removes the DNS domain server from your configuration.

The **dhcpcd enable if\_name** command enables the DHCP daemon to begin to listen for the DHCP client requests on the DHCP-enabled interface. The **no dhcpcd enable** command disables the DHCP server feature on the specified interface.

DHCP must be enabled to use this command. Use the **dhcpcd enable if\_name** command to turn on DHCP.

**Note**

The PIX Firewall DHCP server daemon does not support clients that are not directly connected to a firewall interface, and the interface must be configured to retrieve DHCP client information (with the **dhcpdrelay enable *client\_ifc*** command).

The **dhcpd option 66 | 150** command retrieves TFTP server address information for Cisco IP Phone connections.

When a **dhcpd option** command request arrives at the PIX Firewall DHCP server, the PIX Firewall places the value(s) specified by the **dhcpd option 66 | 150** in the response.

Use the **dhcpd option *code*** command as follows:

- If the TFTP server for Cisco IP Phone connections is located on the inside interface, use the local IP address of the TFTP server in the **dhcpd option** command.
- If the TFTP server is located on a less secure interface, create a group of NAT, **global** and **access-list** command statements for the inside IP phones, and use the actual IP address of the TFTP server in the **dhcpd option** command.
- If the TFTP server is located on a more secure interface, create a group of **static** and **access-list** command statements for the TFTP server and use the global IP address of the TFTP server in the **dhcpd option** command.

The **show dhcpd** command displays **dhcpd** commands, binding and statistics information associated with all of the **dhcpd** commands.

The **clear dhcpd** command clears all of the **dhcpd** commands, binding, and statistics information.

The **debug dhcpd event** command displays event information about the DHCP server. The **debug dhcpd packet** command displays packet information about the DHCP server. Use the **no** form of the **debug dhcpd** commands to disable debugging.

**Examples**

The following partial configuration example shows how to use the **dhcpd address**, **dhcpd dns**, and **dhcpd enable *if\_name*** commands to configure an address pool for the DHCP clients and a DNS server address for the DHCP client, and how to enable the **dmz** interface of the PIX Firewall for the DHCP server function.

```
dhcpd address 10.0.1.100-10.0.1.108 dmz
dhcpd dns 209.165.200.226
dhcpd enable dmz
```

The following partial configuration example shows how to define a DHCP pool of 253 addresses and use the **auto\_config** command to configure the DNS, WINS, and DOMAIN parameters. Note that the **dmz** interface of the firewall is configured as the DHCP server, and the netmask of the **dmz** interface is 255.255.254.0:

```
ip address dmz 10.0.1.1 255.255.254.0
dhcpd address 10.0.1.2-10.0.1.254 dmz
dhcpd auto_config outside
dhcpd enable dmz
```

The following partial configuration example shows how to use three new features that are associated with each other: DHCP server, DHCP client, and PAT using interface IP to configure a PIX Firewall in a small office, home office (SOHO) environment with the **inside** interface as the DHCP server:

```
! use dhcp to configure the outside interface and default route
ip address outside dhcp setroute
! enable dhcp server daemon on the inside interface
ip address inside 10.0.1.2 255.255.255.0
dhcpd address 10.0.1.100-10.0.1.108 inside
dhcpd dns 209.165.201.2 209.165.202.129
dhcpd wins 209.165.201.5
dhcpd lease 3600
dhcpd domain example.com
dhcpd enable inside
! use outside interface IP as PAT global address
nat (inside) 1 0 0
global (outside) 1 interface
```

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

```
pixfirewall(config)# show dhcpd
dhcpd address 10.0.1.100-10.0.1.108 inside
dhcpd lease 3600
dhcpd ping_timeout 750
dhcpd dns 209.165.201.2 209.165.202.129
dhcpd enable inside
```

The following is sample output from the **show dhcpd binding** command:

```
pixfirewall(config)# show dhcpd binding
IP Address Hardware Address Lease Expiration Type
10.0.1.100 0100.a0c9.868e.43 84985 seconds automatic
```

The following is sample output from the **show dhcpd statistics** command:

```
show dhcpd statistics
Address Pools 1
Automatic Bindings 1
Expired Bindings 1
Malformed messages 0

Message Received
BOOTREQUEST 0
DHCPDISCOVER 1
DHCPREQUEST 2
DHCPDECLINE 0
DHCPRELEASE 0
DHCPINFORM 0

Message Sent
BOOTREPLY 0
DHCPOFFER 1
DHCPACK 1
DHCPNAK 1
```

## Related Commands

<b>ip address</b>	Configures the IP address and mask for an interface, or defines a local address pool.
-------------------	---



# dhcprelay

Configures the DHCP relay agent, which relays requests between the firewall interface of the DHCP server and DHCP clients on a different firewall interface.

**[no] dhcprelay enable** *client\_ifc*

**[no] dhcprelay server** *dhcp\_server\_ip server\_ifc*

**[no] dhcprelay setroute** *client\_ifc*

**[no] dhcprelay timeout** *seconds*

**[clear|show] dhcprelay [statistics]**

## Syntax Description

<i>client_ifc</i>	The name of the interface on which the DHCP relay agent accepts client requests.
<i>dhcp_server_ip</i>	The IP address of the DHCP server to which the DHCP relay agent forwards client requests.
<i>enable</i>	Enables the DHCP relay agent to accept DHCP requests from clients on the specified interface.
<i>seconds</i>	The number of seconds allowed for DHCP relay address negotiation.
<i>server_ifc</i>	The name of the firewall interface on which the DHCP server resides.
<i>statistics</i>	The DHCP relay statistics, incremented until a <b>clear dhcprelay statistics</b> command is issued.

## Defaults

By default, the DHCP relay agent is disabled.

The default DHCP relay timeout value is 60 seconds.

## Command Modes

Configuration mode. The **show dhcprelay** commands are also available in privileged mode.

## Usage Guidelines

Use the **dhcprelay enable**, **dhcprelay server**, and **dhcprelay timeout** commands to configure the DHCP relay agent to relay requests between the firewall interface of the DHCP server and DHCP clients on a different firewall interface.



### Note

Use network extension mode for DHCP clients whose DHCP server is on the other side of an Easy VPN tunnel. Otherwise, if the DHCP client is behind a PIX Firewall VPN Easy Remote device connected to an Easy VPN Server using client mode, then the DHCP client will not be able to get a DHCP IP address from the DHCP server on the other side of the Easy VPN Server.

**dhcprelay enable**

For the firewall to start the DHCP relay agent with the **dhcprelay enable** *client\_ifc* command, you must have a **dhcprelay server** command already in your configuration. Otherwise, the firewall displays an error message similar to the following:

```
DHCPRA:Warning - There are no DHCP servers configured!
               No relaying can be done without a server!
               Use the 'dhcprelay server <server_ip> <server_ifc>' command
```

The **dhcprelay enable** *client\_ifc* command starts a DHCP server task on the specified interface. If this **dhcprelay enable** command is the first **dhcprelay enable** command to be issued, and there are **dhcprelay server** commands in the configuration, then the ports for the DHCP servers referenced are opened and the DHCP relay task starts.

When a **dhcprelay enable** *client\_ifc* command is removed with a **no dhcprelay enable** *client\_ifc* command, the DHCP server task for that interface stops. When the **dhcprelay enable** command being removed is the last **dhcprelay enable** command in the configuration, all of the ports for the servers specified in the **dhcprelay server** commands are closed and the DHCP relay task stops.

**dhcprelay server**

Add at least one **dhcprelay server** command to your firewall configuration before you enter a **dhcprelay enable** command or the firewall will issue an error message.

The **dhcprelay server** command opens a UDP port 67 on the specified interface for the specified server and starts the DHCP relay task as soon as a **dhcprelay enable** command is added to the configuration. If there is **no dhcprelay enable** command in the configuration, then the sockets are not opened and the DHCP relay task does not start.

When a **dhcprelay server** *dhcp\_server\_ip* [*server\_ifc*] command is removed, the port for that server is closed. If the **dhcprelay server** command being removed is the last **dhcprelay server** command in the configuration, then the DHCP relay task stops.

**dhcprelay setroute**

The **dhcprelay setroute** *client\_ifc* command enables you to configure the DHCP Relay Agent to change the first default router address (in the packet sent from the DHCP server) to the address of *client\_ifc*. That is, the DHCP Relay Agent substitutes the address of the default router with the address of *client\_ifc*.

If there is no default router option in the packet, the firewall adds one containing the address of *client\_ifc*. This allows the client to set its default route to point to the firewall.

When the **dhcprelay setroute** *client\_ifc* command is not configured (and there is a default router option in the packet) it passes through the firewall with the router address unaltered.

**dhcprelay timeout**

The **dhcprelay timeout** command sets the amount of time, in seconds, allowed for responses from the DHCP server to pass to the DHCP client through the relay binding structure.

**no dhcprelay commands**

The **no dhcprelay enable** *client\_ifc* command removes the DHCP relay agent configuration for the interface specified by *client\_ifc* only.

The **no dhcprelay server** *dhcp\_server\_ip* [*server\_ifc*] command removes the DHCP relay agent configuration for the DHCP server and specified by *dhcp\_server\_ip* [*server\_ifc*] only.

**show dhcprelay**

The **show dhcprelay** command displays the DHCP relay agent configuration, and the **show dhcprelay statistics** command displays counters for the packets relayed by the DHCP relay agent.

The **clear dhcprelay** command clears all DHCP relay configurations. The **clear dhcprelay statistics** command clears the **show dhcprelay statistics** counters.

---

**Examples**

The following example configures the DHCP relay agent for a DHCP server with the IP address of 10.1.1.1 on the outside interface of the firewall and client requests on the inside interface of the firewall, and sets the timeout value to 90 seconds:

```
pixfirewall(config)# dhcprelay server 10.1.1.1 outside
pixfirewall(config)# show dhcprelay
dhcprelay server 10.1.1.1 outside
dhcprelay timeout 50
```

```
pixfirewall(config)# dhcprelay timeout 60
pixfirewall(config)# show dhcprelay
dhcprelay server 10.1.1.1 outside
dhcprelay timeout 60
```

```
pixfirewall(config)# dhcprelay enable inside
pixfirewall(config)# show dhcprelay
dhcprelay server 10.1.1.1 outside
dhcprelay enable inside
dhcprelay timeout 60
```

The following example shows how to disable the DHCP relay agent if there is only one **dhcprelay enable** command in the configuration:

```
pixfirewall(config)# no dhcprelay enable
pixfirewall(config)# show dhcprelay
dhcprelay server 10.1.1.1 outside
dhcprelay timeout 60
```

The following is sample output from the **show dhcprelay statistics** command:

```
pixfirewall(config)# show dhcprelay statistics
Packets Relayed
BOOTREQUEST          0
DHCPDISCOVER         7
DHCPREQUEST          3
DHCPDECLINE          0
DHCPRELEASE          0
DHCPINFORM           0

BOOTREPLY            0
DHCPPOFFER           7
DHCPACK              3
DHCPNAK              0
```

---

**Related Commands**

<a href="#">dhcpd</a>	Controls the DHCP server feature.
-----------------------	-----------------------------------

---

# disable

Exit privileged mode and return to unprivileged mode.

**enable**

**disable**

## Syntax Description

<b>enable</b>	Enter this at the PIX Firewall command-line interface prompt to enter privileged mode.
<b>disable</b>	Enter this at the PIX Firewall command-line interface prompt to exit privileged mode.

## Command Modes

Privileged mode.

## Usage Guidelines

Use the **enable** command to enter privileged mode. The **disable** command exits privileged mode and returns you to unprivileged mode.

## Examples

The following example shows how to enter privileged mode:

```
pixfirewall> enable
pixfirewall#
```

The following example shows how to exit privileged mode:

```
pixfirewall# disable
pixfirewall>
```

# domain-name

Change the IPSec domain name.

**domain-name** *name*

## Syntax Description

<i>name</i>	A domain name, up to 63 characters.
-------------	-------------------------------------

## Command Modes

Configuration mode.

## Usage Guidelines

The **domain-name** command lets you change the IPSec domain name.

**Note**

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

**Examples**

The following example shows use of the **domain-name** command:

```
domain-name example.com
```

## dynamic-map

View or delete a dynamic crypto map entry. To configure crypto dynamic map entries, see the [crypto dynamic-map](#) command.

```
clear dynamic-map
```

```
show dynamic-map
```

**Syntax Description**

dynamic-map	A dynamic crypto map entry.
-------------	-----------------------------

**Command Modes**

Configuration mode.

**Usage Guidelines**

The **clear dynamic-map** command removes **dynamic-map** commands from the configuration. The **show dynamic-map** command lists the **dynamic-map** commands in the configuration.

**Note**

The **dynamic-map** command is the same as the **crypto dynamic-map** command. Refer to the [crypto dynamic-map](#) command page for more information such as examples and other command options.

## EEPROM

Displays and updates the contents of the EEPROM non-volatile storage devices used for low-level Ethernet interface configuration information. This command applies only to Cisco Pix Firewall 506E, 515E, and 525 models.

```
EEPROM update
```

```
show EEPROM
```

**Syntax Description**

update	Restores the contents of the EEPROM registers to a default value. The first three EEPROM registers, which contain MAC address information, are not affected by this command.
--------	--

**Command Modes**

Configuration mode.

**Usage Guidelines**

The **eeprom update** command was added in Version 5.2(4) and can be used to fix corruption of the EEPROM for the onboard Ethernet interfaces of PIX 506E, 515E, and 525 models. Use the **show eeprom** command to display the current EEPROM settings.

The **eeprom update** command verifies the EEPROM register settings and resets them if they are not set to the default values. If the **eeprom update** command updates the EEPROM settings, a reboot of the PIX Firewall is recommended. If the **eeprom update** command does not update the settings a reboot is not recommended.

The **eeprom update** command performs the same function as the **eedisk** utility without requiring access to the ROM monitor mode. The **eeprom update** command does not change the settings of the first three registers, which represent the MAC address of the interface.

The PIX Firewall packet driver does not utilize all of the registers. In addition to the first three registers, the PIX Firewall utilizes the first two bits of Register 3, and all of Register 6. The contents of Register 5, 10, and 12 are ignored by the PIX Firewall packet driver. Each register is 16 bits. The correct register values are shown in [Table 5-2](#):

**Table 5-2** *EEPROM Registers*

Register	Name	Value
Register 0 to 2	MAC address	Differs on each system (unique)
Register 3	Compatibility Bits	0x3 or 0xe03
Register 5	Controller and connector type	0x201*
Register 6	Onboard PHY type	0x4701
Register 10	Onboard Prom ID	0x40C0 or 0x4882*
Register 12	Vendor ID, where 8086 is Intel	0x8086*

\*Ignored by the PIX Firewall packet driver.

**Examples**

The **show eeprom** command displays the current EEPROM register settings, as shown in the following example:

```
pixfirewall# show eeprom
eeprom settings for ifc0:
  reg0: 0x5000
  reg1: 0xfe54
  reg2: 0x65f6
  reg3: 0x3
  reg5: 0x201
  reg6: 0x4702
  reg10: 0x40c0
  reg12: 0x8086
eeprom settings for ifc1:
  reg0: 0x5000
  reg1: 0xfe54
  reg2: 0x66f6
  reg3: 0x3
  reg5: 0x201
  reg6: 0x4702
```

```
reg10: 0x40c0
reg12: 0x8086
```

If you enter the **show eeprom** command on a unit that is not a PIX 506E, 515E, or 525, the following message is displayed:

```
pixfirewall# show eeprom
This unit is not a PIX-525.
Type help or '?' for a list of available commands.
```

If you need to run an update, the **eeprom update** command prompts for a system restart as shown in the following example:

```
pixfirewall# eeprom update
eeprom settings on ifc0 are being reset to defaults:
  reg0: 0x5000
  reg1: 0xfe54
  reg2: 0x65f6
  reg3: 0x3
  reg5: 0x201
  reg6: 0xffff0x4701
  reg10: 0xffff0x40c0
  reg12: 0xffff0x8086
eeprom settings on ifc1 are being reset to defaults:
  reg0: 0x5000
  reg1: 0xfe54
  reg2: 0x66f6
  reg3: 0x3
  reg5: 0x201
  reg6: 0x4701
  reg10: 0x40c0
  reg12: 0x8086
*** WARNING! *** WARNING! *** WARNING! *** WARNING! ***
The system should be restarted as soon as possible.
*** WARNING! *** WARNING! *** WARNING! *** WARNING! ***
```

If the PIX Firewall EEPROM settings are already set to the default, the eeprom update command will not execute and the output will appear as follows:

```
pixfirewall# eeprom update
eeprom settings on ifc0 are already up to date:
  reg0: 0x5000
  reg1: 0xfe54
  reg2: 0x65f6
  reg3: 0x3
  reg5: 0x201
  reg6: 0x4701
  reg10: 0x40c0
  reg12: 0x8086
eeprom settings on ifc1 are already up to date:
  reg0: 0x5000
  reg1: 0xfe54
  reg2: 0x66f6
  reg3: 0x3
  reg5: 0x201
  reg6: 0x4701
  reg10: 0x40c0
  reg12: 0x8086
```

# enable

Start privileged mode or access privilege levels.

```
enable [priv_level]

disable [priv_level]

enable password [pw] [level priv_level] [encrypted]

no enable password [level priv_level]

show enable
```

Syntax Description

<b>enable</b>	Specifies to activate a process, mode, or privilege level.
<b>enable</b> <i>priv_level</i>	Specifies to enable the privilege level, from 0 to 15.
<b>encrypted</b>	Specifies that the provided password is already encrypted.
<b>level</b> <i>priv_level</i>	Specifies to set the privilege level, from 0 to 15.
<b>password</b>	Specifies to configure privilege levels.
<i>pw</i>	The privilege level password string.

Command Modes

Unprivileged mode for **enable**, and configuration mode for **enable password**.

Usage Guidelines

The **enable** command starts privileged mode(s). The PIX Firewall prompts you for your privileged mode password. By default, a password is not required—press the **Enter** key at the Password prompt to start privileged mode. Use the **disable** command to exit privileged mode. Use the **enable password** command to change the password.

The **enable password** command changes the privileged mode password, for which you are prompted after you enter the **enable** command. When the PIX Firewall starts and you enter privileged mode, the password prompt appears. There is not a default password (press the **Enter** key at the Password prompt).

You can return the enable password to its original value (press the **Enter** key at prompt) by entering the following command:

```
pixfirewall# enable password
pixfirewall#
```



Note

If you change the password, write it down and store it in a manner consistent with your site’s security policy. Once you change this password, you cannot view it again. Also, ensure that all who access the PIX Firewall console are given this password.

Use the **passwd** command to set the password for Telnet access to the PIX Firewall console. The default **passwd** value is **cisco**.

See the **passwd** command page for more information.

If no privilege level name is specified, then the highest privilege level is assumed.



The **show enable** command displays the password configuration for privilege levels.

---

**Examples**

The following example shows how to start privileged mode with the **enable** command and then configuration mode with the **configure terminal** command.

```
pixfirewall> enable
Password:
pixfirewall# configure terminal
pixfirewall(config)#
```

The following examples show how to start privileged mode with the **enable** command, change the enable password with the **enable password** command, enter configuration mode with the **configure terminal** command, and display the contents of the current configuration with the **write terminal** command:

```
pixfirewall> enable
Password:
pixfirewall# enable password w0ttallfe
pixfirewall# configure terminal
pixfirewall(config)# write terminal
Building configuration...
...
enable password 2oifudsaoiD.9ff encrypted
...
```

The following example shows the use of the **encrypted** option:

```
enable password 1234567890123456 encrypted
show enable password
enable password 1234567890123456 encrypted

enable password 1234567890123456
show enable password
enable password feCkwUGktTCaGIbD encrypted
```

The following example shows how to configure enable passwords for levels other than the default level of 15:

```

pixfirewall(config)# enable password cisco level 10

pixfirewall(config)# show enable
enable password wC38a.EQklqK3ZqY level 10 encrypted
enable password 8Ry2YjIyt7RRXU24 encrypted

pixfirewall(config)# enable password wC38a.EQklqK3ZqY level 12 encrypted

pixfirewall(config)# show enable
enable password wC38a.EQklqK3ZqY level 10 encrypted
enable password wC38a.EQklqK3ZqY level 12 encrypted
enable password 8Ry2YjIyt7RRXU24 encrypted

pixfirewall(config)# no enable password level 12

pixfirewall(config)# show enable
enable password wC38a.EQklqK3ZqY level 10 encrypted
enable password 8Ry2YjIyt7RRXU24 encrypted

pixfirewall(config)# no enable password level 10

pixfirewall(config)# show enable
enable password 8Ry2YjIyt7RRXU24 encrypted

```

However, notice that defining privilege levels 10 and 12 does not change or remove the level 15 password.

## established

Permit return connections on ports other than those used for the originating connection based on an established connection.

```
[no] established <est_protocol> <dport> [sport] [permitto <protocol> <port>[-<port>]] [permitfrom
<protocol> <port>[-<port>]]
```

**clear established**

**show established**

### Syntax Description

<i>dest_port</i>	Specifies the destination port to use for the established connection lookup. This is the originating traffic's destination port and may be specified as 0 if the protocol does not specify which destination port(s) will be used. Use wildcard ports (0) only when necessary.
<b>permitfrom</b>	Used to specify the return traffic's protocol and from which source port(s) the traffic will be permitted.
<b>permitto</b>	Used to specify the return traffic's protocol and to which destination port(s) the traffic will be permitted.
<i>src_port</i>	Specifies the source port to use for the established connection lookup. This is the originating traffic's source port and may be specified as 0 if the protocol does not specify which source port(s) will be used. Use wildcard ports (0) only when necessary.

**Command Modes**

Configuration mode.

**Usage Guidelines**

The **established** command allows outbound connections return access through the PIX Firewall. This command works with two connections, an original connection outbound from a network protected by the PIX Firewall and a return connection inbound between the same two devices on an external host.

The first protocol, destination port, and optional source port specified are for the initial outbound connection. The **permitto** and **permitfrom** options refine the return inbound connection.

**Note**

We recommend that you always specify the **established** command with the **permitto** and **permitfrom** options. Without these options, the use of the **established** command opens a security hole that can be exploited for attack of your internal systems. See the “Security Problem” section that follows for more information.

The **permitto** option lets you specify a new protocol or port for the return connection at the PIX Firewall.

The **permitfrom** option lets you specify a new protocol or port at the remote server.

The **no established** command disables the **established** feature.

The **clear established** command removes all **establish** command statements from your configuration.

**Note**

For the **established** command to work properly, the client must listen on the port specified with the **permitto** option.

You can use the **established** command with the **nat 0** command statement (where there are no **global** command statements).

**Note**

The **established** command cannot be used with Port Address Translation (PAT).

The **established** command works as shown in the following format:

```
established A B C permitto D E permitfrom D F
```

This command works as though it were written “If there exists a connection between two hosts using protocol A from src port B destined for port C, permit return connections through the PIX Firewall via protocol D (D can be different from A), if the source port(s) correspond to F and the destination port(s) correspond to E.”

For example:

```
established tcp 6060 0 permitto tcp 6061 permitfrom tcp 6059
```

In this case, if a connection is started by an internal host to an external host using TCP source port 6060 and any destination port, the PIX Firewall permits return traffic between the hosts via TCP destination port 6061 and TCP source port 6059.

For example:

```
established udp 0 6060 permitto tcp 6061 permitfrom tcp 1024-65535
```

In this case, if a connection is started by an internal host to an external host using UDP destination port 6060 and any source port, the PIX Firewall permits return traffic between the hosts via TCP destination port 6061 and TCP source port 1024-65535.

### Security Problem

The **established** command has been enhanced to optionally specify the destination port used for connection lookups. Only the source port could be specified previously with the destination port being 0 (a wildcard). This addition allows more control over the command and provides support for protocols where the destination port is known, but the source port is not.

The **established** command can potentially open a large security hole in the PIX Firewall if not used with discretion. Whenever you use this command, if possible, also use the **permitto** and **permitfrom** options to indicate ports to which and from which access is permitted. Without these options, external systems to which connections are made could make unrestricted connections to the internal host involved in the connection. The following are examples of potentially serious security violations that could be allowed when using the **established** command.

For example:

```
established tcp 0 4000
```

In this example, if an internal system makes a TCP connection to an external host on port 4000, then the external host could come back in on any port using any protocol:

```
established tcp 0 0 (Same as previous releases established tcp 0 command.)
```

### Examples

The following example occurs when a local host 10.1.1.1 starts a TCP connection on port 9999 to a foreign host 209.165.201.1. The example allows packets from the foreign host 209.165.201.1 on port 4242 back to local host 10.1.1.1 on port 5454.

```
established tcp 9999 permitto tcp 5454 permitfrom tcp 4242
```

The next example allows packets from foreign host 209.165.201.1 on any port back to local host 10.1.1.1 on port 5454:

```
established tcp 9999 permitto tcp 5454
```

### XDMCP Support

PIX Firewall now provides support for XDMCP (X Display Manager Control Protocol) with assistance from the **established** command.

XDMCP is on by default, but will not complete the session unless the **established** command is used.

For example:

```
established tcp 0 6000 permitto tcp 6000 permitfrom tcp 1024-65535
```

This enables the internal XDMCP equipped (UNIX or ReflectionX) hosts to access external XDMCP equipped XWindows servers. UDP/177 based XDMCP negotiates a TCP based XWindows session and subsequent TCP back connections will be permitted. Because the source port(s) of the return traffic is unknown, the *src\_port* field should be specified as 0 (wildcard). The destination port, *dest\_port*, will typically be 6000; the well-known XServer port. The *dest\_port* should be 6000 + *n*; where *n* represents the local display number. Use the following UNIX command to change this value.

```
setenv DISPLAY hostname:displaynumber.screennumber
```

The **established** command is needed because many TCP connections are generated (based on user interaction) and the source port for these connection is unknown. Only the destination port will be static. The PIX Firewall does XDMCP fixups transparently. No configuration is required, but the **established** command is necessary to accommodate the TCP session. Be advised that using applications like this through the PIX Firewall may open up security holes. The XWindows system has been exploited in the past and newly introduced exploits are likely to be discovered.

# exit

Exit an access mode.

**exit**

**enable**

## Syntax Description

<b>exit</b>	Exits the current command mode.
<b>enable</b>	Enables privileged mode.

## Command Modes

All modes.

## Usage Guidelines

Use the **exit** command to exit from an access mode. This command is the same as the **quit** command.

## Examples

The following example shows how to exit configuration mode and then privileged mode:

```
pixfirewall(config)# exit
pixfirewall# exit
pixfirewall>
```

# failover

Enable or disable the PIX Firewall failover feature on a standby PIX Firewall.

**[no] failover [active]**

**[no] failover ip address** *if\_name ip\_address*

**[no] failover lan unit** **primary** | **secondary**

**[no] failover lan interface** *lan\_if\_name*

**[no] failover lan key** *key\_secret*

**[no] failover lan enable**

**[no] failover link** [*stateful\_if\_name*]

**[no] failover mac address** *mif\_name act\_mac stn\_mac*

**[no] failover poll** *seconds*

**[no] failover replicate** **http**

**failover reset****show failover** [**lan** [**detail**]]

Syntax	Description
<i>act_mac</i>	The interface MAC address for the active PIX Firewall.
<b>active</b>	Make a PIX Firewall the active unit. Use this command when you need to force control of the connection back to the unit you are accessing, such as when you want to switch control back from a unit after you have fixed a problem and want to restore service to the primary unit. Either enter the <b>no failover active</b> command on the secondary unit to switch service to the primary or the <b>failover active</b> command on the primary unit.
<i>detail</i>	Displays LAN-based failover configuration information.
<i>enable</i>	Enables LAN-based failover; otherwise, serial cable failover is used.
<i>if_name</i>	The interface name for the failover IP address.
<i>ip_address</i>	The IP address used by the standby unit to communicate with the active unit. Use this IP address with the <b>ping</b> command to check the status of the standby unit. This address must be on the same network as the system IP address. For example, if the system IP address is 192.159.1.3, set the failover IP address to 192.159.1.4.
<i>key</i>	Enables encryption and authentication of LAN-based failover messages between PIX Firewalls.
<i>key_secret</i>	The shared secret key.
<i>lan</i>	Specifies LAN-based failover.
<i>lan interface</i> <i>lan_if_name</i>	The name of the firewall interface dedicated to LAN-based failover. The interface name of a VLAN logical interface cannot be used for <i>lan_if_name</i> .
<b>link</b>	Specify the interface where a Fast Ethernet or Gigabit LAN link is available for Stateful Failover. A VLAN logical interface cannot be used.
<i>mif_name</i>	The name of the interface to set the MAC address.
<b>poll seconds</b>	Specify how long failover waits before sending special failover “hello” packets between the primary and standby units over all network interfaces and the failover cable. The default is 15 seconds. The minimum value is 3 seconds and the maximum is 15 seconds. Set to a lower value for Stateful Failover. With a faster poll time, PIX Firewall can detect failure and trigger failover faster. However, faster detection may cause unnecessary switchovers when the network is temporarily congested or a network card starts slowly.
<i>primary</i>	Specifies the primary PIX Firewall to use for LAN-based failover.
<i>replicate http</i>	The <b>[no] failover replicate http</b> command allows the stateful replication of HTTP sessions in a Stateful Failover environment. The <b>no</b> form of this command disables HTTP replication in a Stateful Failover configuration. When HTTP replication is enabled, the <b>show failover</b> command displays the <b>failover replicate http</b> command configuration.
<b>reset</b>	Force both units back to an unfailed state. Use this command once the fault has been corrected. The <b>failover reset</b> command can be entered from either unit, but it is best to always enter commands at the active unit. Entering the <b>failover reset</b> command at the active unit will “unfail” the standby unit.
<i>secondary</i>	Specifies the secondary PIX Firewall to use for LAN-based failover.

<i>stateful_if_name</i>	In addition to the failover cable, a dedicated Fast Ethernet or Gigabit LAN link is required to support Stateful Failover. The interface name of a VLAN logical interface cannot be used for <i>stateful_if_name</i> .
<i>stn_mac</i>	The interface MAC address for the standby PIX Firewall.

**Command Modes**

Configuration mode.

**Usage Guidelines**

The default failover setup uses serial cable failover. LAN-based failover requires explicit LAN-based failover configuration. Additionally, for LAN-based failover, you must install a dedicated 100 Mbps or Gigabit Ethernet, full-duplex VLAN switch connection for failover operations. Failover is not supported using a crossover Ethernet cable between two PIX Firewall units.

**Note**

The PIX 506/506E cannot be used for failover in any configuration.

The primary unit in the PIX 515/515E, PIX 525, or PIX 535 failover pair must have an Unrestricted (UR) license. The secondary unit can have Failover (FO) or UR license. However, the failover pair must be two otherwise identical units with the same PIX Firewall hardware and software.

For a Stateful Failover link, use the **mtu** command to set the interface maximum transmission unit (MTU) to 1500 bytes or greater.

For serial cable failover, use the **failover** command without an argument after you connect the optional failover cable between your primary PIX Firewall and a secondary PIX Firewall. The default configuration has failover enabled. Enter **no failover** in the configuration file for the PIX Firewall if you will not be using the failover feature. Use the **show failover** command to verify the status of the connection and to determine which unit is active.

For LAN-based failover, use the **failover lan** commands. The **show failover lan** command displays LAN-based failover information (only), and **show failover lan detail** supplies debugging information for your LAN-based failover configuration.

**Note**

Refer to the *Cisco PIX Firewall and VPN Configuration Guide* for configuration information.

For failover, the PIX Firewall requires that you configure any unused interfaces with one of the following methods:

- Shutdown the interface and do not configure its IP or failover IP address. If these addresses are configured, use the **no ip address** and **no failover ip address** commands to remove the configuration.
- Configure the interface like other interfaces but use a cross-over Ethernet cable to connect the interface to the Standby unit. Do not connect the interface to an external switch or hub device.

Set the speed of the Stateful Failover dedicated interface to 100full for a Fast Ethernet interface or 1000fullsx for a Gigabit Ethernet interface.

Use the **failover active** command to initiate a failover switch from the standby unit, or the **no failover active** command from the active unit to initiate a failover switch. You can use this feature to return a failed unit to service, or to force an active unit off line for maintenance. Because the standby unit does not keep state information on each connection, all active connections will be dropped and must be re-established by the clients.

Use the **failover link** command to enable Stateful Failover. Enter the **no failover link** command to disable the Stateful Failover feature.

If a failover IP address has not been entered, the **show failover** command will display 0.0.0.0 for the IP address, and monitoring of the interfaces will remain in “waiting” state. A failover IP address must be set for failover to work.

The **failover mac address** command enables you to configure a virtual MAC address for a PIX Firewall failover pair. The **failover mac address** command sets the PIX Firewall to use the virtual MAC address stored in the PIX Firewall configuration after failover, instead of obtaining a MAC address by contacting its failover peer. This enables the PIX Firewall failover pair to maintain the correct MAC addresses after failover. If a virtual MAC address is not specified, the PIX Firewall failover pair uses the burned in network interface card (NIC) address as the MAC address. However, the **failover mac address** command is unnecessary (and therefore cannot be used) on an interface configured for LAN-based failover because the **failover lan interface lan\_if\_name** command does not change the IP and MAC addresses when failover occurs.

When adding the **failover mac address** command to your configuration, it is best to configure the virtual MAC address, save the configuration to Flash memory, and then reload the PIX Firewall pair. If the virtual MAC address is added when there are active connections, then those connections will stop. Also, you must write the complete PIX Firewall configuration, including the **failover mac address** command, into the Flash memory of the secondary PIX Firewall for the virtual MAC addressing to take effect.

The **failover poll seconds** command lets you determine how long failover waits before sending special failover “hello” packets between the primary and standby units over all network interfaces and the failover cable. The default is 15 seconds. The minimum value is 3 seconds and the maximum is 15 seconds. Set to a lower value for Stateful Failover. With a faster poll time, PIX Firewall can detect failure and trigger failover faster. However, faster detection may cause unnecessary switchovers when the network is temporarily congested or a network card starts slowly.

When a failover cable connects two PIX Firewall units, the **no failover** command now disables failover until you enter the **failover** command to explicitly enable failover. Previously, when the failover cable connected two PIX Firewall units and you entered the **no failover** command, failover would automatically re-enable after 15 seconds.

You can also view the information from the **show failover** command using SNMP. Refer to the *Cisco PIX Firewall and VPN Configuration Guide* for more information on configuring failover.

### Usage Notes

1. LAN-based failover requires a dedicated interface, but the same interface can also be used for Stateful Failover. However, the interface needs enough capacity to handle both the LAN-based failover and Stateful Failover traffic; otherwise, use two separate dedicated interfaces.
2. If you reboot the PIX Firewall without entering the **write memory** command and the failover cable is connected, failover mode automatically enables.

## Examples

### Serial Cable (Default) Failover

The following sample output shows that failover is enabled, and that the primary unit state is active:

```
show failover
pixfirewall (config)# show failover
  Failover On
  Cable status:Normal
  Reconnect timeout 0:00:00
  Poll frequency 15 seconds
  Last Failover at: 18:32:16 UTC Mon Apr 7 2003
  failover replication http
```



```

This host:Secondary - Standby
  Active time:0 (sec)
  Interface FailLink (209.165.201.6):Normal
  Interface 4th (209.165.200.230):Normal
  Interface int5 (209.165.200.226):Normal
  Interface intf2 (192.168.1.1):Normal
  Interface outside (209.165.200.225):Normal
  Interface inside (10.1.1.4):Normal
Other host:Primary - Active
  Active time:242145 (sec)
  Interface FailLink (172.16.31.1):Normal

```

The rest of command output is omitted.

The “Cable status” has these values:

- Normal—Indicates that the active unit is working and that the standby unit is ready.
- Waiting—Indicates that monitoring of the other unit’s network interfaces has not yet started.
- Failed—Indicates that the PIX Firewall has failed.

The “Stateful Obj” has these values:

- Xmit—Indicates the number of packets transmitted.
- Xerr—Indicates the number of transmit errors.
- Rcv—Indicates the number of packets received.
- Rcv—Indicates the number of receive errors.

Each row is for a particular object static count:

- General—The sum of all stateful objects.
- Sys cmd—Refers to logical update system commands, such as **login** or **stay alive**.
- Up time—The value for PIX Firewall up time which the active PIX Firewall unit will pass on to the standby unit.
- Xlate—The PIX Firewall translation information.
- Tcp conn—The PIX Firewall dynamic TCP connection information.
- Udp conn—The PIX Firewall dynamic UDP connection information.
- ARP tbl—The PIX Firewall dynamic ARP table information.
- RIF tbl—The dynamic router table information.

The Standby Logical Update Statistics output displayed when you use the **show failover** command only describes Stateful Failover. The “xerrs” value does not indicate an error in failover, but rather the number of packet transmit errors.

You can view the IP addresses of the standby unit with the **show ip address** command:

```

show ip address
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.

### LAN-Based Failover

To make sure LAN-based failover starts properly, follow these configuration steps:

- 
- Step 1** Configure the primary PIX Firewall unit before connecting the failover LAN interface.
  - Step 2** Save the primary unit configuration to Flash memory.
  - Step 3** Configure the PIX Firewall secondary unit using the appropriate **failover lan** commands before connecting the LAN-based failover interface.
  - Step 4** Save the secondary unit configuration to Flash memory.
  - Step 5** Reboot both units and connect the LAN-based failover interfaces to the designated failover switch, hub, or VLAN.
  - Step 6** If any item in a **failover lan** command needs to be changed, then disconnect the LAN-based failover interface, and repeat the preceeding steps.
- 



#### Note

When properly configured, the LAN-based failover configurations for your primary and secondary PIX Firewall units should be different, reflecting which is primary and which is secondary.

The following example outlines how to configure LAN-based failover between two PIX Firewall units.

#### Primary PIX Firewall configuration:

```

:
pix(config)# nameif ethernet0 outside security0
pix(config)# nameif ethernet1 inside security100
pix(config)# nameif ethernet2 stateful security20
pix(config)# nameif ethernet3 lanlink security30
:
pix(config)#interface ethernet0 100full
pix(config)#interface ethernet1 100full
pix(config)#interface ethernet2 100full
pix(config)#interface ethernet3 100full
pix(config)#interface ethernet4 100full
:
pix(config)# ip address outside 172.23.58.70 255.255.255.0
pix(config)# ip address inside 10.0.0.2 255.255.255.0
pix(config)# ip address stateful 10.0.1.2 255.255.255.0
pix(config)# ip address lanlink 10.0.2.2 255.255.255.0
pix(config)# failover ip address outside 172.23.58.51
pix(config)# failover ip address inside 10.0.0.4
pix(config)# failover ip address stateful 10.0.1.4
pix(config)# failover ip address lanlink 10.0.2.4
pix(config)# failover
pix(config)# failover poll 15
pix(config)# failover lan unit primary
pix(config)# failover lan interface lanlink
pix(config)# failover lan key 12345678
pix(config)# failover lan enable
:

```

#### Secondary PIX Firewall configuration:

```

Pix2(config)# nameif ethernet3 lanlink security30
pix2(config)# interface ethernet3 100full

```

```

pix2(config)# ip address lanlink 10.0.2.2 255.255.255.0
pix2(config)# failover ip address lanlink 10.0.2.4
pix2(config)# failover
pix2(config)# failover lan unit secondary (optional)
pix2(config)# failover lan interface lanlink
pix2(config)# failover lan key 12345678
pix2(config)# failover lan enable

```

The following example illustrates how to use the **failover mac address** command:

```

ip address outside 172.23.58.50 255.255.255.224
ip address inside 192.168.2.11 255.255.255.0
ip address intf2 192.168.10.11 255.255.255.0
failover
failover ip address outside 172.23.58.51
failover ip address inside 192.168.2.12
failover ip address intf2 192.168.10.12
failover mac address outside 00a0.c989.e481 00a0.c969.c7f1
failover mac address inside 00a0.c976.cde5 00a0.c922.9176
failover mac address intf2 00a0.c969.87c8 00a0.c918.95d8
failover link intf2
...:

```

The output of the **show failover** command includes a section for LAN-based failover if it is enabled as follows:

```

pix(config)# show failover
Failover On
Cable status: Unknown
Reconnect timeout 0:00:00
Poll frequency 15 seconds
Last Failover at: 18:32:16 UTC Mon Apr 7 2003
    This host: Primary - Standby
        Active time: 255 (sec)
        Interface outside (192.168.1.232): Normal
        Interface inside (192.168.5.2): Normal
    Other host: Secondary - Active
        Active time: 256305 (sec)
        Interface outside (192.168.1.231): Normal
        Interface inside (192.168.5.1): Normal

Stateful Failover Logical Update Statistics
    Link : Unconfigured.

Lan Based Failover is Active
    interface dmz (209.165.200.226): Normal, peer (209.165.201.1): Normal

```

The **show failover lan** command displays only the LAN-based failover section, as follows:

```

pix(config)# show failover lan
Lan Based Failover is Active
    interface dmz (209.165.200.226): Normal, peer (209.165.201.1): Normal

```

The **show failover lan detail** command is used mainly for debugging purposes and displays information similar to the following:

```
pix(config)# show failover lan detail
Lan Failover is Active
This Pix is Primary
Command Interface is dmz
Peer Command Interface IP is 209.165.201.1
My interface status is 0x1
Peer interface status is 0x1
Peer interface downtime is 0x0
Total msg send: 103093, rcvd: 103031, dropped: 0, retrans: 13, send_err: 0
Total/Cur/Max of 51486:0:5 msgs on retransQ
...
LAN FO cmd queue, count: 0, head: 0x0, tail: 0x0
Failover config state is 0x5c
Failover config poll cnt is 0
Failover pending tx msg cnt is 0
Failover Fmsg cnt is 0
:
```

## filter

Enable, disable, or view URL, FTP, HTTPS, Java, and ActiveX filtering

**[no] filter activex** *port* | **except** *local\_ip mask foreign\_ip mask*

**[no] filter ftp** *dest-port* | **except** *local\_ip local\_mask foreign\_ip foreign\_mask* [**allow**]  
[**interact-block**]

**[no] filter java** *port*[-*port*]| **except** *local\_ip mask foreign\_ip mask*

**[no] filter https** *dest-port* | **except** *local\_ip local\_mask foreign\_ip foreign\_mask* [**allow**]

**[no] filter url** [**http** | *port*[-*port*]] **except** *local\_ip local\_mask foreign\_ip foreign\_mask* [**allow**]  
[**proxy-block**] [**longurl-truncate** | **longurl-deny**] [**cgi-truncate**]

**[no] filter url except** *local\_ip local\_mask foreign\_ip foreign\_mask*

**[no] filter url port** | **except** *local\_ip mask foreign\_ip mask* [**allow**] [**proxy-block**]  
[**longurl-truncate** | **longurl-deny**] [**cgi-truncate**]

**clear filter**

**show filter**

Syntax	Description
<b>activex</b>	Block inbound ActiveX, and other HTML <object> tags from outbound packets.
<b>allow</b>	<b>filter url</b> only: When the server is unavailable, let outbound connections pass through the firewall without filtering. If you omit this option, and if the N2H2 or Websense server goes off line, PIX Firewall stops outbound port 80 (Web) traffic until the N2H2 or Websense server is back on line.
<b>cgi_truncate</b>	Sends a CGI script as an URL.
<i>dest-port</i>	The destination port number.

<b>except</b>	Creates an exception to a previously specified set of IP addresses.
<i>foreign_ip</i>	The IP address of the lowest security level interface to which access is sought. You can use <b>0.0.0.0</b> (or in shortened form, <b>0</b> ) to specify all hosts.
<i>foreign_mask</i>	Network mask of <i>foreign_ip</i> . Always specify a specific mask value. You can use <b>0.0.0.0</b> (or in shortened form, <b>0</b> ) to specify all hosts.
<b>ftp</b>	Enables File Transfer Protocol (FTP) filtering. Available with Websense URL filtering only.
<b>http</b>	Specifies port 80. You can enter <b>http</b> or <b>www</b> instead of 80 to specify port 80.)
<b>https</b>	Enables HTTPS filtering. Available with Websense URL filtering only.
<b>interact-block</b>	Prevents users from connecting to the FTP server through an interactive FTP program.
<b>java</b>	Specifies to filter out Java applets returning from an outbound connection.
<i>local_ip</i>	The IP address of the highest security level interface from which access is sought. You can set this address to <b>0.0.0.0</b> (or in shortened form, <b>0</b> ) to specify all hosts.
<i>local_mask</i>	Network mask of <i>local_ip</i> . You can use <b>0.0.0.0</b> (or in shortened form, <b>0</b> ) to specify all hosts.
<b>longurl-deny</b>	Denies the URL request if the URL is over the URL buffer size limit or the URL buffer is not available.
<i>longurl-truncate</i>	Sends only the originating host name or IP address to the Websense server if the URL is over the URL buffer limit.
<i>mask</i>	Any mask.
<i>port</i>	The port that receives Internet traffic on the PIX Firewall. Typically, this is port 80, but other values are accepted. The <b>http</b> or <b>url</b> literal can be used for port 80.
<b>proxy-block</b>	Prevents users from connecting to an HTTP proxy server.
<b>url</b>	Filter Universal Resource Locators (URLs) from data moving through the PIX Firewall.

**Command Modes**

Configuration mode.

**Usage Guidelines**The **clear filter** command removes all **filter** commands from the configuration.**Note**

The PIX Firewall 501 platform supports a maximum of 15 filter entries.

**filter activex**

The **filter activex** command filters out ActiveX, Java applets, and other HTML <object> usages from outbound packets. ActiveX controls, formerly known as OLE or OCX controls, are components you can insert in a web page or other application. These controls include custom forms, calendars, or any of the extensive third-party forms for gathering or displaying information.

As a technology, it creates many potential problems for the network clients including causing workstations to fail, introducing network security problems, or be used to attack servers.

This feature blocks the HTML <object> tag and comments it out within the HTML web page.

**Note**

The `<object>` tag is also used for Java applets, image files, and multimedia objects, which will also be blocked by the **filter activex** command. If the `<object>` or `</object>` HTML tags split across network packets or if the code in the tags is longer than the number of bytes in the MTU, the PIX Firewall cannot block the tag.

ActiveX blocking does not occur when users access an IP address referenced by the **alias** command.

To specify that all outbound connections have ActiveX blocking, use the following command:

```
filter activex 80 0 0 0 0
```

This command specifies that the ActiveX blocking applies to Web traffic on port 80 from any local host and for connections to any foreign host.

**filter java**

The **filter java** command filters out Java applets that return to the PIX Firewall from an outbound connection. The user still receives the HTML page, but the web page source for the applet is commented out so that the applet cannot execute. Use 0 for the *local\_ip* or *foreign\_ip* IP addresses to mean all hosts.

**Note**

If Java applets are known to be in `<object>` tags, use the **filter activex** command to remove them.

To specify that all outbound connections have Java applet blocking, use the following command:

```
filter java 80 0 0 0 0
```

This command specifies that the Java applet blocking applies to Web traffic on port 80 from any local host and for connections to any foreign host.

**filter url**

The **filter url** command lets you prevent outbound users from accessing World Wide Web URLs that you designate using the N2H2 or Websense filtering application.

**Note**

The **url-server** command must be configured before issuing the **filter** command for HTTPS and FTP, and if all URL servers are removed from the server list, then all **filter** commands related to URL filtering are also removed.

The **allow** option to the **filter** command determines how the PIX Firewall behaves in the event that the N2H2 or Websense server goes off line. If you use the **allow** option with the **filter** command and the N2H2 or Websense server goes offline, port 80 traffic passes through the PIX Firewall without filtering. Used without the **allow** option and with the server off line, PIX Firewall stops outbound port 80 (Web) traffic until the server is back on line, or if another URL server is available, passes control to the next URL server.

**Note**

With the **allow** option set, PIX Firewall now passes control to an alternate server if the N2H2 or Websense server goes off line.

The N2H2 or Websense server works with the PIX Firewall to deny users from access to websites based on the company security policy.

Websense protocol Version 4 enables group and username authentication between a host and a PIX Firewall. The PIX Firewall performs a username lookup, and then Websense server handles URL filtering and username logging.

The N2H2 server must be a Windows workstation (2000, NT, or XP), running an IFP Server, with a recommended minimum of 512 MB of RAM. Also, the long URL support for the N2H2 service is capped at 3 KB, less than the cap for Websense.

Websense protocol Version 4 contains the following enhancements:

- URL filtering allows the PIX Firewall to check outgoing URL requests against the policy defined on the Websense server.
- Username logging tracks username, group, and domain name on the Websense server.
- Username lookup enables the PIX Firewall to use the user authentication table to map the host's IP address to the username.

Follow these steps to filter URLs:

- 
- |               |   |
|---------------|---|
| <b>Step 1</b> | Designate an N2H2 or Websense server with the appropriate vendor-specific form of the <b>url-server</b> command.  |
| <b>Step 2</b> | Enable filtering with the <b>filter</b> command.  |
| <b>Step 3</b> | If needed, improve throughput with the <b>url-cache</b> command. However, this command does not update Websense logs, which may affect Websense accounting reports. Accumulate Websense run logs before using the <b>url-cache</b> command. |
| <b>Step 4</b> | Use the <b>show url-cache stats</b> and the <b>show perfmon</b> commands to view run information.   |
- 

Information on Websense is available at the following website:

<http://www.websense.com/>

### Examples

The following example filters all outbound HTTP connections except those from the 10.0.2.54 host:

```
url-server (perimeter) host 10.0.1.1
filter url 80 0 0 0 0
filter url except 10.0.2.54 255.255.255.255 0 0
```

The following example blocks all outbound HTTP connections destined to a proxy server that listens on port 8080:

```
filter url 8080 0 0 0 0 proxy-block
```

## fixup protocol

Modifies PIX Firewall protocol fixups to add, delete, or change services and feature defaults.

**fixup protocol ctique 2748**

**[no] fixup protocol dns [maximum-length *length*]**

**fixup protocol esp-ike**

```

fixup protocol ftp [strict] [port]

fixup protocol h323 {h225 | ras} port [-port]

fixup protocol http [port[-port]]

fixup protocol icmp error

fixup protocol ils [port[-port]]

[no] fixup protocol mgcp [port[-port]]

fixup protocol pptp 1723

fixup protocol rsh [514]

fixup protocol rtsp [port]

fixup protocol sip [port[-port]]

[no] fixup protocol sip udp 5060

fixup protocol skinny [port[-port]]

fixup protocol smtp [port[-port]]

fixup protocol snmp [161[-162]]

fixup protocol sqlnet [port[-port]]

fixup protocol tftp [port[-port]]

no fixup protocol [protocol_name] [port]

clear fixup

show etiqbe

show fixup

show fixup protocol protocol [protocol]

show conn state [protocol]

show h225

show h245

show h323-ras

show mgcp

show sip

show skinny

show timeout protocol

```



Syntax	Description
<b>ctiqbe</b>	Enables the Computer Telephony Interface Quick Buffer Encoding (CTIQBE) fixup. Used with Cisco TAPI/JTAPI applications.
<b>dns</b>	Enables the DNS fixup.
<b>esp-ike</b>	Enables PAT for Encapsulating Security Payload (ESP), single tunnel.
<b>fixup protocol ils</b>	Provides support for Microsoft NetMeeting, SiteServer, and Active Directory products that use LDAP to exchange directory information with an ILS server.
<b>fixup protocol</b> <i>protocol</i> [ <i>protocol</i> ] <i>[port[-port]]</i>	Modifies PIX Firewall protocol fixups to add, delete, or change services and feature defaults.
<b>ftp</b>	Specifies to change the ftp port number.
<b>h323 h225</b>	Specifies to use H.225, the ITU standard that governs H.225.0 session establishment and packetization, with H.323. H.225.0 actually describes several different protocols: RAS, use of Q.931, and use of RTP.
<b>h323 ras</b>	Specifies to use RAS with H.323 to enable dissimilar communication devices to communicate with each other. H.323 defines a common set of CODECs, call setup and negotiating procedures, and basic data transport methods.
<b>http</b> [ <i>port</i> [- <i>port</i> ]]	The default port for HTTP is 80. Use the <i>port</i> option to change the HTTP port, or the <i>port-port</i> option to specify a range of HTTP ports.
<b>ils</b>	Specifies the Internet Locator Service. The default port is TCP LDAP server port 389.
<b>dns</b> <b>maximum-length</b> <i>length</i>	Specifies the maximum DNS packet length allowed. Default is 512 bytes.
<b>mgcp</b>	Enables the Media Gateway Control Protocol (MGCP) fixup. (Use the <b>mgcp</b> command to configure additional support for the MGCP fixup.)
<b>no</b>	Disables the fixup of a protocol by removing all fixups of the protocol from the configuration using the <b>no fixup</b> command. After removing all fixups for a protocol, the <b>no fixup</b> form of the command or the default port is stored in the configuration.
<i>port</i>	The port on which to enable the fixup (application inspection). You can use port numbers or supported port literals. The default ports are: TCP 21 for <b>ftp</b> , TCP LDAP server port 389 for <b>ils</b> , TCP 80 for <b>http</b> , TCP 1720 for <b>h323 h225</b> , UDP 1718-1719 for <b>h323 ras</b> , TCP 514 for <b>rsh</b> , TCP 554 for <b>rtsp</b> , TCP 2000 for <b>skinny</b> , TCP 25 for <b>smtp</b> , TCP 1521 for <b>sqlnet</b> , TCP 5060 for <b>sip</b> , and UDP 69 for TFTP. The default port value for <b>rsh</b> cannot be changed, but additional port statements can be added. See the “Ports” section in <a href="#">Chapter 2, “Using PIX Firewall Commands”</a> for a list of valid port literal names. The port over which the designated protocol travels.
<i>port-port</i>	Specifies a port range.
<b>pptp</b>	Enables Point-to-Point Tunneling Protocol (PPTP) application inspection. The default port is 1723.
<i>protocol</i>	Specifies the protocol to fix up.
<i>protocol_name</i>	The protocol name.
<b>ras</b>	Registration, admission, and status (RAS) is a signaling protocol that performs registration, admissions, bandwidth changes, status, and disengage procedures between the VoIP gateway and the gatekeeper.

<b>rsh</b>	The Remote Shell (RSH) protocol uses a TCP connection from the RSH client to the RSH server on TCP port 514. The client and server negotiate the TCP port number where the client will listen for the STDERR output stream. RSH inspection supports NAT of the negotiated port number if necessary.
<b>sip</b>	Enable or change the port assignment for the Session Initiation Protocol (SIP) for Voice over IP TCP connections. UDP SIP is on by default and can be disabled and the port assignment is nonconfigurable. PIX Firewall Version 6.2 introduced PAT support for SIP.
<b>skinny</b>	Enable SCCP application inspection. The default port is <b>2000</b> . SCCP protocol supports IP telephony and can coexist in an H.323 environment. An application layer ensures that all SCCP signaling and media packets can traverse the PIX Firewall and interoperate with H.323 terminals.  Skinny is the short name form for Skinny Client Control Protocol (SCCP).
<b>snmp</b>	Disabled by default. Enables SNMP inspection if enabled.
<b>sqlnet</b>	Changes the default port assignment for Oracle SQL*Net.
<b>strict</b>	Prevent web browsers from sending embedded commands in FTP requests. Each FTP command must be acknowledged before a new command is allowed. Connections sending embedded commands are dropped.
<b>tftp</b>	Enable TFTP application inspection. The default port is <b>69</b> .
<b>upd</b>	Specifies the UDP port number.

### Command Modes

All **fixup protocol** commands are available in configuration mode unless otherwise specified.  
The **show fixup protocol mgcp** command is available in privileged mode.

### Defaults

The default ports for the PIX Firewall fixup protocols are as follows:

```

pixHA(config)# sh fix
fixup protocol dns maximum-length 512
fixup protocol ftp 21
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol http 80
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol sip 5060
fixup protocol sip udp 5060
fixup protocol skinny 2000
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol tftp 69
pixHA(config)#

```

(These are the defaults enabled on a PIX Firewall running software Version 6.3(4).)

The fixup for MGCP is disabled by default.

## Usage Guidelines

The **fixup protocol** commands let you view, change, enable, or disable the use of a service or protocol through the PIX Firewall. The ports you specify are those that the PIX Firewall listens at for each respective service. You can change the port value for every service except **rsh**. The **fixup protocol** commands are always present in the configuration and are enabled by default.

The **fixup protocol** command performs the Adaptive Security Algorithm based on different port numbers other than the defaults. This command is global and changes things for both inbound and outbound connections, and cannot be restricted to any **static** command statements.

The **clear fixup** command resets the fixup configuration to its default. It does not remove the default **fixup protocol** commands.

You can disable the fixup of a protocol by removing all fixups of the protocol from the configuration using the **no fixup** command. After you remove all fixups for a protocol, the **no fixup** form of the command or the default port is stored in the configuration.

### show fixup commands

The **show fixup** command displays the current fixup configuration and port values.

The **show fixup protocol protocol [protocol]** command displays the port values for the individual protocol specified.

The **show conn state [sip]** command displays the connection state of the designated protocol.

The **show h225**, **show h245**, and **show h323-ras** commands display connection information for troubleshooting H.323 fixup issues, and are described with the **fixup protocol h323 {h225 | ras}** commands.

The **show skinny** command assists in troubleshooting SKINNY fixup issues and is described with the **fixup protocol skinny** command.

The **show sip** command assists in troubleshooting SIP fixup issues and is described with the **fixup protocol sip udp 5060** command. The **show timeout sip** command displays the timeout value of the designated protocol.

### fixup protocol ctique 2748

The **fixup protocol ctique 2748** command enables CTIQBE protocol inspection that supports NAT, PAT, and bi-directional NAT. This enables Cisco IP SoftPhone and other Cisco TAPI/JTAPI applications to work successfully with Cisco CallManager for call setup across the firewall.

By default, **fixup protocol ctique 2748** is disabled. You enable the CTIQBE fixup as shown in the following example:

```
pixfirewall(config)# fixup protocol ctique 2748
```

```
pixfirewall(config)# show fixup protocol ctique  
fixup protocol ctique 2748
```

The **no fixup protocol ctique 2748** command disables the CTIQBE fixup.

The **show ctique** command displays information of CTIQBE sessions established across the PIX Firewall. Along with **debug ctique** and **show local-host**, this command is used for troubleshooting CTIQBE fixup issues.



#### Note

We recommend that you have the **pager** command configured before using the **show ctique** command. If there are a lot of CTIQBE sessions and the **pager** command is not configured, it can take a while for the **show ctique** command output to reach the end.

The following is sample output from the **show ctiqbe** command under the following conditions. There is only one active CTIQBE session setup across the PIX Firewall. It is established between an internal CTI device (for example, a Cisco IP SoftPhone) at local address 10.0.0.99 and an external Cisco Call Manager at 172.29.1.77, where TCP port 2748 is the Cisco CallManager. The heartbeat interval for the session is 120 seconds.

```
pixfirewall(config)# show ctiqbe
```

```
Total: 1
      LOCAL          FOREIGN          STATE    HEARTBEAT
-----
1      10.0.0.99/1117  172.29.1.77/2748      1         120
-----
      RTP/RTCP: PAT xlates: mapped to 172.29.1.99(1028 - 1029)
-----
      MEDIA: Device ID 27      Call ID 0
              Foreign 172.29.1.99      (1028 - 1029)
              Local   172.29.1.88      (26822 - 26823)
-----
```

The CTI device has already registered with the CallManager. The device's internal address and RTP listening port is PATed to 172.29.1.99 UDP port 1028. Its RTCP listening port is PATed to UDP 1029.

The line beginning with **RTP/RTCP: PAT xlates:** appears only if an internal CTI device has registered with an external CallManager and the CTI device's address and ports are PATed to that external interface. This line does not appear if the CallManager is located on an internal interface, or if the internal CTI device's address and ports are NATed to the same external interface that is used by the CallManager.

The output indicates a call has been established between this CTI device and another phone at 172.29.1.88. The RTP and RTCP listening ports of the other phone are UDP 26822 and 26823. The other phone locates on the same interface as the CallManager because the PIX Firewall does not maintain a CTIQBE session record associated with the second phone and CallManager. The active call leg on the CTI device side can be identified with Device ID 27 and Call ID 0.

The following is the xlate information for these CTIBQE connections:

```
pixfirewall(config)# show xlate debug
3 in use, 3 most used
Flags: D - DNS, d - dump, I - identity, i - inside, n - no random,
       o - outside, r - portmap, s - static
TCP PAT from inside:10.0.0.99/1117 to outside:172.29.1.99/1025 flags ri idle 0:00:22
timeout 0:00:30
UDP PAT from inside:10.0.0.99/16908 to outside:172.29.1.99/1028 flags ri idle 0:00:00
timeout 0:04:10
UDP PAT from inside:10.0.0.99/16909 to outside:172.29.1.99/1029 flags ri idle 0:00:23
timeout 0:04:10
```

### fixup protocol dns

Use the **fixup protocol dns** command to specify the maximum DNS packet length. DNS requires application inspection so that DNS queries are not subject to the generic UDP handling based on activity timeouts. Instead, UDP connections associated with DNS queries and responses are torn down as soon as a reply to a DNS query has been received.

The port assignment for the Domain Name System (DNS) is not configurable.

Set the maximum length for the DNS fixup as shown in the following example:

```
pixfirewall(config)# fixup protocol dns maximum-length 1500

pixfirewall(config)# show fixup protocol dns
fixup protocol dns maximum length 1500
```

**Note**

The PIX Firewall drops DNS packets sent to UDP port 53 that are larger than the configured maximum length. The default value is 512 bytes. A syslog message will be generated when a DNS packet is dropped.

The **no fixup protocol dns** command disables the DNS fixup. The **clear fixup protocol dns** resets the DNS fixup to its default settings (512 byte maximum packet length).

**Note**

If the DNS fixup is disabled, the A-record is not NATed and the DNS ID is not matched in requests and responses. By disabling the DNS fixup, the maximum length check on UDP DNS packets can be bypassed and packets greater than the maximum length configured will be permitted.

**fixup protocol esp-ike**

The **fixup protocol esp-ike** command enables PAT for Encapsulating Security Payload (ESP), single tunnel.

The **fixup protocol esp-ike** command is disabled by default. If a **fixup protocol esp-ike** command is issued, the fixup is turned on, and the firewall preserves the source port of the Internet Key Exchange (IKE) and creates a PAT translation for ESP traffic. Additionally, if the **esp-ike** fixup is on, ISAKMP cannot be turned on any interface.

**fixup protocol ftp**

Use the **fixup protocol ftp** command to specify the listening port or ports for the File Transfer Protocol (FTP). The following list describes the features and usage of this command:

- The PIX Firewall listens to port 21 for FTP by default.
- Multiple ports can be specified.
- Only specify the port for the FTP control connection and not the data connection. The PIX Firewall stateful inspection will dynamically prepare the data connection as necessary. For example, the following is incorrect:

*INCORRECT*

```
fixup protocol ftp 21
fixup protocol ftp 20
```

and the following is correct:

*CORRECT*

```
fixup protocol ftp 21
```

- Use caution when moving FTP to a higher port. For example, if you set the FTP port to 2021 by entering **fixup protocol ftp 2021** all connections that initiate to port 2021 will have their data payload interpreted as FTP commands.

The following is an example of a **fixup protocol ftp** command configuration that uses multiple FTP fixups:

```
:
: For a PIX Firewall with two interfaces
:
ip address outside 192.168.1.1 255.255.255.0
ip address inside 10.1.1.1 255.255.255.0
:
: There is an inside host 10.1.1.15 that will be
: exported as 192.168.1.15. This host runs the FTP
: services at port 21 and 1021
:
static (inside, outside) 192.168.1.15 10.1.1.15
:
: Construct an access list to permit inbound FTP traffic to
: port 21 and 1021
:
access-list outside permit tcp any host 192.168.1.15 eq ftp
access-list outside permit tcp any host 192.168.1.15 eq 1021
access-group outside in interface outside
:
: Specify that traffic to port 21 and 1021 are FTP traffic
:
fixup protocol ftp 21
fixup protocol ftp 1021
```

If you disable FTP fixups with the **no fixup protocol ftp** command, outbound users can start connections only in passive mode, and all inbound FTP is disabled.

The **strict** option in the **fixup protocol ftp** command performs two separate functions:

- The **strict** option prevents web browsers from sending embedded commands in FTP requests. Each FTP command must be acknowledged before a new command is allowed. Connections sending embedded commands are dropped. The **strict** option only lets an FTP server generate the 227 command and only lets an FTP client generate the PORT command. The 227 and PORT commands are checked to ensure they do not appear in an error string.
- The **strict** option also prevents the PIX from opening up return connections based solely on the information sent in the PORT command. The **strict** option enables the PIX to make sure a successful reply is sent from the server in addition to the PORT command before opening the connection. If an error is seen, the PORT command is ignored by the PIX and the connection is never established. This keeps the PIX from opening data connections for communication that will never occur.

#### **fixup protocol h323 {h225 | ras}**

The **fixup protocol h323 {h225 | ras}** command provides support for H.323 compliant applications such as Cisco CallManager and VocalTec Gatekeeper. H.323 is a suite of protocols defined by the International Telecommunication Union (ITU) for multimedia conferences over LANs.

PIX Firewall software Version 5.3 and higher supports H.323 v2 with Fast Connect or Fast Start Procedure for faster call setup and H.245 tunneling for resource conservation, call synchronization, and reduced set up time.

PIX Firewall software Versions 6.2 and higher support PAT for H.323. When upgrading from any pre-PIX Firewall software Version 6.2 release, the following will be added to the configuration:

```
fixup protocol h323 ras 1718-1719
```

Additionally, **fixup protocol h323 port** becomes **fixup protocol h323 h225 port**. You can disable H.225 signaling or RAS fixup (or both) with the **no fixup protocol h323 {h225 | ras} port [-port]** command.

PIX Firewall software Version 6.3 and higher supports H.323 v3 and v4 messages as well as the H.323 v3 feature Multiple Calls on One Call Signaling Channel.

The **show h225**, **show h245**, and **show h323-ras** commands display connection information for troubleshooting H.323 fixup issues.



#### Note

Before using the **show h225**, **show h245**, or **show h323-ras** commands, we recommend that you configure the **pager** command. If there are a lot of session records and the **pager** command is not configured, it may take a while for the **show** output to reach its end. If there is an abnormally large number of connections, check that the sessions are timing out based on the default timeout values or the values set by you. If they are not, then there is a problem that needs to be investigated.

The **show h225** command displays information for H.225 sessions established across the PIX Firewall. Along with the **debug h323 h225 event**, **debug h323 h245 event**, and **show local-host** commands, this command is used for troubleshooting H.323 fixup issues.

The following is sample output from the **show h225** command:

```
pixfirewall(config)# show h225
Total H.323 Calls: 1
1 Concurrent Call(s) for
  Local: 10.130.56.3/1040 Foreign: 172.30.254.203/1720
  1. CRV 9861
  Local: 10.130.56.3/1040 Foreign: 172.30.254.203/1720
0 Concurrent Call(s) for
  Local: 10.130.56.4/1050 Foreign: 172.30.254.205/1720
```

This output indicates that there is currently 1 active H.323 call going through the PIX Firewall between the local endpoint 10.130.56.3 and foreign host 172.30.254.203, and for these particular endpoints, there is 1 concurrent call between them, with a CRV (Call Reference Value) for that call of 9861.

For the local endpoint 10.130.56.4 and foreign host 172.30.254.205, there are 0 concurrent Calls. This means that there is no active call between the endpoints even though the H.225 session still exists. This could happen if, at the time of the **show h225** command, the call has already ended but the H.225 session has not yet been deleted. Alternately, it could mean that the two endpoints still have a TCP connection opened between them because they set “maintainConnection” to TRUE, so the session is kept open until they set it to FALSE again, or until the session times out based on the H.225 timeout value in your configuration.

The **show h245** command displays information for H.245 sessions established across the PIX Firewall by endpoints using slow start. (Slow start is when the two endpoints of a call open another TCP control channel for H.245. Fast start is where the H.245 messages are exchanged as part of the H.225 messages on the H.225 control channel.) Along with the **debug h323 h245 event**, **debug h323 h225 event**, and **show local-host** commands, this command is used for troubleshooting H.323 fixup issues.

The following is sample output from the **show h245** command:

```
pixfirewall(config)# show h245
Total: 1
LOCAL          TPKT    FOREIGN          TPKT
1 10.130.56.3/1041 0 172.30.254.203/1245 0
MEDIA: LCN 258 Foreign 172.30.254.203 RTP 49608 RTCP 49609
      Local 10.130.56.3 RTP 49608 RTCP 49609
MEDIA: LCN 259 Foreign 172.30.254.203 RTP 49606 RTCP 49607
      Local 10.130.56.3 RTP 49606 RTCP 49607
```

There is currently one H.245 control session active across the PIX Firewall. The local endpoint is 10.130.56.3, and we are expecting the next packet from this endpoint to have a TPKT header since the TPKT value is 0. (The TPKT header is a 4-byte header preceding each H.225/H.245 message. It gives

the length of the message, including the 4-byte header.) The foreign host endpoint is 172.30.254.203, and we are expecting the next packet from this endpoint to have a TPKT header since the TPKT value is 0.

The media negotiated between these endpoints have a LCN (logical channel number) of 258 with the foreign RTP IP address/port pair of 172.30.254.203/49608 and a RTCP IP address/port of 172.30.254.203/49609 with a local RTP IP address/port pair of 10.130.56.3/49608 and a RTCP port of 49609.

The second LCN of 259 has a foreign RTP IP address/port pair of 172.30.254.203/49606 and a RTCP IP address/port pair of 172.30.254.203/49607 with a local RTP IP address/port pair of 10.130.56.3/49606 and RTCP port of 49607.

The **show h323-ras** command displays information for H.323 RAS sessions established across the PIX Firewall between a gatekeeper and its H.323 endpoint. Along with the **debug h323 ras event** and **show local-host** commands, this command is used for troubleshooting H.323 RAS fixup issues.

The following is sample output from the **show h323-ras** command:

```
pixfirewall(config)# show h323-ras
Total: 1
      GK                      Caller
      172.30.254.214 10.130.56.14
```

This output shows that there is one active registration between the gatekeeper 172.30.254.214 and its client 10.130.56.14.

#### fixup protocol http

The **fixup protocol http** command sets the port for Hypertext Transfer Protocol (HTTP) traffic application inspection. The default port for HTTP is 80.

Use the *port* option to change the default port assignments from 80. Use the *port-port* option to apply HTTP application inspection to a range of port numbers.



#### Note

The **no fixup protocol http** command still enables the **filter url** command.

HTTP inspection performs several functions:

- URL logging of GET messages
- URL screening through N2H2 or Websense
- Java and ActiveX filtering

The latter two features must be configured in conjunction with the **filter** command.

#### fixup protocol icmp error

The **fixup protocol icmp error** command enables NAT of ICMP error messages. This creates translations for intermediate hops based on the static or network address translation configuration on the firewall.

The **no fixup protocol icmp error** disables the creation of a translation (xlate) for the intermediate nodes which generate ICMP error messages.

By default **fixup protocol icmp error** is disabled.



### fixup protocol ils

The **fixup protocol ils** command provides NAT support for Microsoft NetMeeting, SiteServer, and Active Directory products that use LightWeight Directory Access Protocol (LDAP) to exchange directory information with an Internet Locator Service (ILS) server.

By default, **fixup protocol ils** is disabled. You enable the ILS fixup as shown in the following example:

```
pixfirewall(config)# fixup protocol ils
```

The **no fixup protocol ils** command disables the ILS fixup.

### fixup protocol mgcp

If a user wishes to use MGCP, they will usually need to configure at least two **fixup protocol** commands: one for the port on which the gateway receives commands, and one for the port on which the Call Agent receives commands.

Normally, a Call Agent sends commands to the default MGCP port for gateways, 2427, and a gateway sends commands to the default MGCP port for Call Agents, 2727.

The following example adds fixup support for Call Agents and gateways that use the default ports:

```
fixup protocol mgcp 2427  
fixup protocol mgcp 2727
```

The **no fixup protocol mgcp** command removes the MGCP fixup configuration.

The **show fixup protocol mgcp** command displays the configured MGCP fixups. Please refer to the **mgcp** command for information on the **show mgcp** command.

### fixup protocol pptp

The **fixup protocol pptp 1723** command inspects PPTP protocol packets and dynamically creates the GRE connections and xlates necessary to permit PPTP traffic.

Specifically, the firewall inspects the PPTP version announcements and the outgoing call request/response sequence. Only PPTP Version 1, as defined in RFC 2637, is inspected. Further inspection on the TCP control channel is disabled if the version announced by either side is not Version 1. In addition, the outgoing-call request and reply sequence are tracked. Connections and/or xlates are dynamically allocated as necessary to permit subsequent secondary GRE data traffic.

The **fixup protocol pptp 1723** command is disabled by default. Enter the following command to enable the PPTP fixup:

```
pixfirewall(config)# fixup protocol pptp 1723  
pixfirewall(config)# show fixup  
fixup protocol ftp 21  
fixup protocol http 80  
fixup protocol h323 h225 1720  
fixup protocol h323 ras 1718-1719  
fixup protocol ils 389  
fixup protocol rsh 514  
fixup protocol rtsp 554  
fixup protocol smtp 25  
fixup protocol sqlnet 1521  
fixup protocol sip 5060  
fixup protocol skinny 2000  
fixup protocol pptp 1723  
fixup protocol sip udp 5060  
fixup protocol tftp 69
```

The PPTP fixup must be enabled for PPTP traffic to be translated by PAT. Additionally, PAT is only performed for a modified version of GRE (RFC2637) and only if it is negotiated over the PPTP TCP control channel. PAT is not performed for the unmodified version of GRE (RFC 1701 and RFC 1702).

### fixup protocol rsh

The RSH protocol uses a TCP connection from the RSH client to the RSH server on TCP port 514. The client and server negotiate the TCP port number where the client will listen for the STDERR output stream. RSH inspection supports NAT of the negotiated port number if necessary.

### fixup protocol rtsp

The **fixup protocol rtsp** command lets PIX Firewall pass Real Time Streaming Protocol (RTSP) packets. RTSP is used by RealAudio, RealNetworks, Apple QuickTime 4, RealPlayer, and Cisco IP/TV connections.

If you are using Cisco IP/TV, use RTSP TCP port 554 and TCP 8554:

```
fixup protocol rtsp 554
fixup protocol rtsp 8554
```

The following restrictions apply to the **fixup protocol rtsp** command:

1. This PIX Firewall will not fix RTSP messages passing through UDP ports.
2. PAT is not supported with the **fixup protocol rtsp** command.
3. PIX Firewall does not have the ability to recognize HTTP cloaking where RTSP messages are hidden in the HTTP messages.
4. PIX Firewall cannot perform NAT on RTSP messages because the embedded IP addresses are contained in the SDP files as part of HTTP or RTSP messages. Packets could be fragmented and PIX Firewall cannot perform NAT on fragmented packets.
5. With Cisco IP/TV, the number of NATs the PIX Firewall performs on the SDP part of the message is proportional to the number of program listings in the Content Manager (each program listing can have at least six embedded IP addresses).
6. You can configure NAT for Apple QuickTime 4 or RealPlayer. Cisco IP/TV only works with NAT if the Viewer and Content Manager are on the outside network and the server is on the inside network.
7. When using RealPlayer, it is important to properly configure transport mode. For the PIX Firewall, add an **access-list** command statement from the server to the client or vice versa. For RealPlayer, change transport mode by clicking **Options>Preferences>Transport>RTSP Settings**.

If using TCP mode on the RealPlayer, select the **Use TCP to Connect to Server** and **Attempt to use TCP for all content** check boxes. On the PIX Firewall, there is no need to configure the fixup.

If using UDP mode on the RealPlayer, select the **Use TCP to Connect to Server** and **Attempt to use UDP for static content** check boxes, and for live content not available via Multicast. On the PIX Firewall, add a **fixup protocol rtsp port** command statement.

### fixup protocol sip

Use the **fixup protocol sip [port[-port]]** command to enable SIP-over-TCP application inspection, or the **fixup protocol sip udp 5060** command to enable SIP-over-UDP application inspection. If either SIP fixup method is enabled, SIP packets are inspected and then NAT is provided for the appropriate IP addresses. The SIP fixups are enabled by default on TCP or UDP port 5060, respectively. However, only

the TCP SIP fixup port is configurable in PIX Firewall software Version 6.3. You cannot change ports on the firewall for the SIP-over-UDP fixup, but you can disable the SIP-over-UDP fixup with the **no fixup protocol sip udp 5060** command.

Session Initiation Protocol (SIP), as defined by the Internet Engineering Task Force (IETF), enables call handling sessions and two-party audio conferences (calls). SIP works with Session Description Protocol (SDP) for call signaling. SDP specifies the ports for the media stream. Using SIP, the PIX Firewall can support any SIP Voice over IP (VoIP) gateway or VoIP proxy server. SIP and SDP are defined in the following RFCs:

- SIP: Session Initiation Protocol, RFC 2543
- SDP: Session Description Protocol, RFC 2327

To support SIP, the following must be inspected: calls through the PIX Firewall, signaling messages for the media connection addresses, media ports, and embryonic connections for the media. This is because while the signaling is sent over a well known destination port (UDP/TCP 5060), the media streams are dynamically allocated and because SIP is a text-based protocol that contains IP addresses throughout the text.

PIX Firewall software Version 6.2 and higher supports PAT for SIP. In PIX Firewall software Version 6.3 and later, you can disable the SIP fixup for both UDP and TCP signaling with the commands **no fixup protocol sip udp 5060** and **no fixup protocol sip [port[-port]]** respectively.

For additional information about the SIP protocol see RFC 2543. For additional information about the Session Description Protocol (SDP), see RFC 2327.

The **show sip** command displays information for SIP sessions established across the PIX Firewall. Along with the **debug sip** and **show local-host** commands, this command is used for troubleshooting SIP fixup issues.



#### Note

We recommend that you configure the **pager** command before using the **show sip** command. If there are a lot of SIP session records and the **pager** command is not configured, it will take a while for the **show sip** command output to reach its end.

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

```
pixfirewall(config)# show sip
Total: 2
call-id c3943000-960ca-2e43-228f@10.130.56.44
    state Call init, idle 0:00:01
call-id c3943000-860ca-7e1f-11f7@10.130.56.45
    state Active, idle 0:00:06
```

This sample shows two active SIP sessions on the PIX Firewall (as shown in the `Total` field). Each `call-id` represents a call.

The first session, with the `call-id c3943000-960ca-2e43-228f@10.130.56.44`, is in the state `Call Init`, which means the session is still in call setup. Call setup is not complete until a final response to the call has been received. For instance, the caller has already sent the INVITE, and maybe received a 100 Response, but has not yet seen the 200 OK, so the call setup is not complete yet. Any non-1xx response message is considered a final response. This session has been idle for 1 second.

The second session is in the state `Active`, in which call setup is complete and the endpoints are exchanging media. This session has been idle for 6 seconds.

**fixup protocol skinny**

Skinny Client Control Protocol (SCCP or “skinny”) protocol supports IP telephony and can coexist in an H.323 environment. An application layer ensures that all SCCP signaling and media packets can traverse the PIX Firewall and interoperate with H.323 terminals. The skinny fixup supports both NAT and PAT configurations.

**Note**

If the address of an internal Cisco CallManager is configured for NAT or PAT to a different IP address or port, registrations for external Cisco IP Phones will fail because the PIX Firewall currently does not support NAT or PAT for the file content transferred via TFTP. Although the PIX Firewall does support NAT of TFTP messages, and opens a pinhole for the TFTP file to traverse the firewall, the PIX Firewall cannot translate the Cisco CallManager IP address and port embedded in the Cisco IP Phone's configuration files that are being transferred using TFTP during phone registration.

If skinny messages are fragmented, then the firewall does not recognize or inspect them. Skinny message fragmentation can occur when a call is established that includes a conference bridge. The firewall tracks the skinny protocol to open conduits for RTP traffic to flow through, however, with the skinny messages fragmented, the firewall cannot correctly set up this conduit.

The **show skinny** command displays information of Skinny (SCCP) sessions established across the PIX Firewall. Along with **debug skinny** and **show local-host**, this command is used for troubleshooting Skinny fixup issues.

**Note**

We recommend that you have the **pager** command configured before using the **show skinny** command. If there are a lot of Skinny sessions and the **pager** command is not configured, it can take a while for the **show skinny** command output to reach the end.

The following is sample output from the **show skinny** command under the following conditions. There are two active Skinny sessions set up across the PIX Firewall. The first one is established between an internal Cisco IP Phone at local address 10.0.0.11 and an external Cisco CallManager at 172.18.1.33. TCP port 2000 is the CallManager. The second one is established between another internal Cisco IP Phone at local address 10.0.0.22 and the same Cisco CallManager.

```
pixfirewall(config)# show skinny
```

	LOCAL	FOREIGN	STATE
1	10.0.0.11/52238	172.18.1.33/2000	1
	MEDIA 10.0.0.11/22948	172.18.1.22/20798	
2	10.0.0.22/52232	172.18.1.33/2000	1
	MEDIA 10.0.0.22/20798	172.18.1.11/22948	

The output indicates a call has been established between both internal Cisco IP Phones. The RTP listening ports of the first and second phones are UDP 22948 and 20798 respectively.

The following is the xlate information for these Skinny connections:

```
pixfirewall(config)# show xlate debug
2 in use, 2 most used
Flags: D - DNS, d - dump, I - identity, i - inside, n - no random,
       o - outside, r - portmap, s - static
NAT from inside:10.0.0.11 to outside:172.18.1.11 flags si idle 0:00:16 timeout 0:05:00
NAT from inside:10.0.0.22 to outside:172.18.1.22 flags si idle 0:00:14 timeout 0:05:00
```

### fixup protocol smtp

The **fixup protocol smtp** command enables the Mail Guard feature, which only lets mail servers receive the RFC 821, section 4.5.1, commands of HELO, MAIL, RCPT, DATA, RSET, NOOP, and QUIT. All other commands are translated into X's which are rejected by the internal server. This results in a message such as “500 Command unknown: 'XXX'.” Incomplete commands are discarded.



#### Note

During an interactive SMTP session, various SMTP security rules may reject or deadlock your Telnet session. These rules include the following: SMTP commands must be at least four characters in length; must be terminated with carriage return and line feed; and must wait for a response before issuing the next reply.

As of PIX Firewall software Version 5.1 and higher, the **fixup protocol smtp** command changes the characters in the SMTP banner to asterisks except for the “2”, “0”, “0 ” characters. Carriage return (CR) and linefeed (LF) characters are ignored.

In PIX Firewall software Version 4.4, all characters in the SMTP banner are converted to asterisks.

### fixup protocol snmp

This snmp fixup command **fixup protocol snmp 161-162** is disabled by default. This command provides the ability to configure a drop of SNMP packets based on protocol version.

The **no fixup protocol snmp** command removes the SNMP fixup configuration.

Fixup can be enabled or disabled with the following command:

**[no] fixup protocol snmp 161-162**



#### Note

Existing connections will retain present fixup configurations from their initial creation.

So, if you toggle the configuration, you need to either:

- Wait for the connections to time out
- Manually clear the connections

Use **clear xlate** or **clear local** to clear connections for the fixup configuration to take effect.

### fixup protocol sqlnet

Use the *port* option to change the default port assignment from 1521. This is the value used by Oracle for SQL\*Net, but this value does not agree with IANA port assignments for Structured Query Language (SQL). Use the *-port* option to apply SQL\*Net inspection to a range of port numbers.

### fixup protocol tftp

PIX Firewall Version 6.3(2) introduced application inspection for Trivial File Transfer Protocol (TFTP). The default port is 69. Use the *port-port* option to apply TFTP application inspection to a range of port numbers.

The PIX Firewall inspects TFTP traffic and dynamically creates connections and translations, if necessary, to permit file transfer between a TFTP client and server with the **fixup protocol tftp** command. Specifically, the fixup inspects TFTP read request (RRQ), write request (WRQ), and error notification (ERROR).

A dynamic secondary channel and a PAT translation, if necessary, are allocated on a reception of a valid read (RRQ) or write (WRQ) request. This secondary channel is subsequently used by TFTP for file transfer or error notification.

Only the TFTP server can initiate traffic over the secondary channel, and at most one incomplete secondary channel can exist between the TFTP client and server. An error notification from the server closes the secondary channel.

The `show fixup protocol tftp` command displays the ports on which TFTP is inspected.

```
pixdoc515(config)# show fixup protocol tftp
fixup protocol tftp 69
```

## Examples

The following example enables access to an inside server running Mail Guard:

```
static (inside,outside) 209.165.201.1 192.168.42.1 netmask 255.255.255.255
access-list acl_out permit tcp any host 209.165.201.1 eq smtp
access-group acl_out in interface outside
fixup protocol smtp 25
```

The following example shows the commands to disable Mail Guard:

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

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

The following example shows how to enable the MGCP fixup on your firewall:

```
pixfirewall(config)# fixup protocol mgcp 2427
pixfirewall(config)# fixup protocol mgcp 2727
pixfirewall(config)# show running-config
: Saved
:
PIX Version 6.3
interface ethernet0 auto
interface ethernet1 auto
interface ethernet2 auto shutdown
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 intf2 security10
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname pixfirewall
domain-name cisco.com
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol ils 389
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
fixup protocol mgcp 2427
fixup protocol mgcp 2727
fixup protocol sip udp 5060
fixup protocol tftp 69
```

```

names
access-list 101 permit tcp any host 10.1.1.3 eq www
access-list 101 permit tcp any host 10.1.1.3 eq smtp
pager lines 24
mtu outside 1500
mtu inside 1500
mtu intf2 1500
ip address outside 172.23.59.232 255.255.0.0
ip address inside 10.1.1.1 255.255.255.0
ip address intf2 192.168.10.12 255.255.255.255
ip audit info action alarm
ip audit attack action alarm
pdm history enable
arp timeout 14400
global (outside) 1 interface
nat (inside) 1 0.0.0.0 0.0.0.0 0 0
routing interface inside
route outside 0.0.0.0 0.0.0.0 172.23.59.225 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h225 1:00:00
timeout h323 0:05:00 mgcp 0:05:00 sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
aaa-server LOCAL protocol local
http server enable
http 10.1.1.2 255.255.255.255 inside
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
floodguard enable
telnet timeout 5
ssh timeout 5
console timeout 0
dhcprelay server 10.1.1.1 outside
terminal width 80
Cryptochecksum:00000000000000000000000000000000
: end

```

The following example shows how to remove the MGCP fixup from your configuration:

```

pixfirewall(config)# show fixup protocol mgcp
fixup protocol mgcp 2427
fixup protocol mgcp 2727
pixfirewall(config)# no fixup protocol mgcp
pixfirewall(config)#

```

#### Related Commands

<b>debug</b>	Displays debug information for application traffic.
<b>mgcp</b>	Configures additional support for the Media Gateway Control Protocol fixup (packet application inspection) and is used with the <b>fixup protocol mgcp</b> command.
<b>show conn</b>	Displays all active connections.
<b>timeout</b>	Sets the maximum idle time duration.

# flashfs

Clear, display, or downgrade filesystem information.

**flashfs downgrade** {**4.x** | **5.0** | **5.1**}

**clear flashfs**

**show flashfs**

## Syntax Description

<b>downgrade 4.x</b>	Clear the filesystem information from Flash memory before downgrading to PIX Firewall software Version 4.0, 4.1, 4.2, 4.3, or 4.4.
<b>downgrade 5.0</b>   <b>5.1</b>	Write the filesystem to Flash memory before downgrading to the appropriate PIX Firewall software Version 5.0 or higher.

## Command Modes

Configuration mode.

## Usage Guidelines

The **clear flashfs** and the **flashfs downgrade 4.x** commands clear the filesystem part of Flash memory in the PIX Firewall. Versions 4.*n* cannot use the information in the filesystem; it needs to be cleared to let the earlier version operate correctly.

The **flashfs downgrade 5.x** command reorganizes the filesystem part of Flash memory so that information stored in the filesystem can be accessed by the earlier version. The PIX Firewall maintains a filesystem in Flash memory to store system information, IPSec private keys, certificates, and CRLs. It is crucial that you clear or reformat the filesystem before downgrading to a previous PIX Firewall version. Otherwise, your filesystem will get out of sync with the actual contents of the Flash memory and cause problems when the unit is later upgraded.



### Note

When downgrading to PIX Firewall Versions 5.0 or 5.1, which support a maximum 4 MB of Flash memory, configuration files larger than 4 MB will be truncated and some configuration information will be lost.

You only need to use the **flashfs downgrade 5.x** command if your PIX Firewall has 16 MB of Flash memory, if you have IPSec private keys, certificates, or CRLs stored in Flash memory, and you used the **ca save all** command to save these items in Flash memory. The **flashfs downgrade 5.x** command fails if the filesystem indicates that any part of the image, configuration, or private data in the Flash memory device is unusable.

The **clear flashfs** and **flashfs downgrade** commands do not affect the configuration stored in Flash memory.

The **clear flashfs** command is the same as the **flashfs downgrade 4.x** command.

The **show flashfs** command displays the size in bytes of each filesystem sector and the current state of the filesystem. The data in each sector is as follows:

- file 0: PIX Firewall binary image, where the .bin file is stored.
- file 1: PIX Firewall configuration data that you can view with the **show config** command
- file 2: PIX Firewall datafile that stores IPSec key and certificate information



- file 3: PDM image
- file 4: crashdump file
- file 5: Filesystem record

### Examples

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

```
pixfirewall(config)# show flashfs
flash file system:  version:3  magic:0x12345679
  file 0: origin:      0 length:2109496
  file 1: origin: 2621440 length:1883
  file 2: origin: 2752512 length:72
  file 3: origin: 2883584 length:3126944
  file 4: origin:      0 length:0
  file 5: origin: 8257536 length:308
pixfirewall(config)#
```

The origin values are integer multiples of the underlying filesystem sector size.

## floodguard

Enable or disable Flood Guard to protect against flood attacks.

**floodguard enable**

**floodguard disable**

**clear floodguard**

**show floodguard**

### Syntax Description

<b>enable</b>	Enable Flood Guard.
<b>disable</b>	Disable Flood Guard.

### Command Modes

Configuration mode.

### Usage Guidelines

The **floodguard** command lets you reclaim PIX Firewall resources if the user authentication (uauth) subsystem runs out of resources. If an inbound or outbound uauth connection is being attacked or overused, the PIX Firewall will actively reclaim TCP user resources.

When the resources deplete, the PIX Firewall lists messages about it being out of resources or out of tcpusers.

If the PIX Firewall uauth subsystem is depleted, TCP user resources in different states are reclaimed depending on urgency in the following order:

1. Timewait
2. LastAck
3. FinWait

4. Embryonic
5. Idle

The **floodguard** command is enabled by default.

---

**Examples**

The following example enables the **floodguard** command and lists the **floodguard** command statement in the configuration:

```
floodguard enable
show floodguard
floodguard enable
```

# fragment

The **fragment** command provides additional management of packet fragmentation and improves compatibility with NFS.

**fragment size** *database-limit* [*interface*]

**fragment chain** *chain-limit* [*interface*]

**fragment timeout** *seconds* [*interface*]

**clear fragment**

**show fragment** [*interface*]

## Syntax Description

<b>chain</b>	Specifies the maximum number of packets into which a full IP packet can be fragmented. The default is 24.
<i>chain-limit</i>	The default is 24. The maximum is 8200.
<b>clear</b>	Resets the fragment databases and defaults. All fragments currently waiting for reassembly are discarded and the <b>size</b> , <b>chain</b> , and <b>timeout</b> options are reset to their default values.
<i>database-limit</i>	The default is 200. The maximum is 1,000,000 or the total number of blocks.
<i>interface</i>	The PIX Firewall interface. If not specified, the command will apply to all interfaces.
<i>seconds</i>	The default is 5 seconds. The maximum is 30 seconds.
<b>show</b>	<ul style="list-style-type: none"> <li>Displays the state of the fragment database:</li> <li>Size—Maximum packets set by the <b>size</b> option.</li> <li>Chain—Maximum fragments for a single packet set by the <b>chain</b> option.</li> <li>Timeout—Maximum seconds set by the <b>timeout</b> option.</li> <li>Queue—Number of packets currently awaiting reassembly.</li> <li>Assemble—Number of packets successfully reassembled.</li> <li>Fail—Number of packets which failed to be reassembled.</li> <li>Overflow—Number of packets which overflowed the fragment database.</li> </ul>
<b>size</b>	Sets the maximum number of packets in the fragment database. The default is 200.
<b>timeout</b>	Specifies the maximum number of seconds that a packet fragment will wait to be reassembled after the first fragment is received before being discarded. The default is 5 seconds.

## Command Modes

Configuration mode.

## Usage Guidelines

By default the PIX Firewall accepts up to 24 fragments to reconstruct a full IP packet. Based on your network security policy, you should consider configuring the PIX Firewall to prevent fragmented packets from traversing the firewall by entering the **fragment chain 1 interface** command on each interface. Setting the limit to 1 means that all packets must be whole; that is, unfragmented.

If a large percentage of the network traffic through the PIX Firewall is NFS, additional tuning may be necessary to avoid database overflow. See system log message 209003 for additional information.

In an environment where the MTU between the NFS server and client is small, such as a WAN interface, the **chain** option may require additional tuning. In this case, NFS over TCP is highly recommended to improve efficiency.

Setting the *database-limit* of the **size** option to a large value can make the PIX Firewall more vulnerable to a DoS attack by fragment flooding. Do not set the *database-limit* equal to or greater than the total number of blocks in the 1550 or 16384 pool. See the **show block** command for more details. The default values will limit DoS due to fragment flooding to that interface only.

The **show fragment [interface]** command displays the states of the fragment databases. If the interface name is specified, only displays information for the database residing at the specified interface.

## Examples

For example, to prevent fragmented packets on the outside and inside interfaces enter the following commands:

```
pixfirewall(config)# fragment chain 1 outside
pixfirewall(config)# fragment chain 1 inside
```

Continue entering the **fragment chain 1 interface** command for each additional interface on which you want to prevent fragmented packets.

The following example configures the outside fragment database to limit a maximum size of 2000, a maximum chain length of 45, and a wait time of 10 seconds:

```
pixfirewall(config)#
pixfirewall(config)# fragment outside size 2000
pixfirewall(config)# fragment chain 45 outside
pixfirewall(config)# fragment outside timeout 10
pixfirewall(config)#
```

The **clear fragment** command resets the fragment databases. Specifically, all fragments awaiting re-assembly are discarded. In addition, the size is reset to 200; the chain limit is reset to 24; and the timeout is reset to 5 seconds.

The **show fragment** command display the states of the fragment databases. If the interface name is specified, only the database residing at the specified interface is displayed.

```
pixfirewall(config)# show fragment outside
Interface:outside
Size:2000, Chain:45, Timeout:10
Queue:1060, Assemble:809, Fail:0, Overflow:0
```


The preceding example shows that the "outside" fragment database has the following:

- A database size limit of 2000 packets.
- The chain length limit of 45 fragments.
- A timeout of ten seconds.
- 1060 packets is currently awaiting re-assembly.
- 809 packets has been fully reassembled.

- No failure.
- No overflow.

This fragment database is under heavy usage.

The PIX Firewall also includes FragGuard for additional IP fragmentation protection. For more information refer to the *Cisco PIX Firewall and VPN Configuration Guide*.

 fragment