



Configuring Application Inspection (Fixup)

This chapter describes how to use and configure application inspection, which is often called “fixup” because you use the **fixup** command to configure it. This chapter includes the following sections:

- [How Application Inspection Works](#)
- [Using the fixup Command](#)
- [Basic Internet Protocols](#)
- [Voice Over IP](#)
- [Multimedia Applications](#)
- [Database and Directory Support](#)
- [Management Protocols](#)

How Application Inspection Works

The Adaptive Security Algorithm (ASA), used by the PIX Firewall for stateful application inspection, ensures the secure use of applications and services. Some applications require special handling by the PIX Firewall application inspection function. Applications that require special application inspection functions are those that embed IP addressing information in the user data packet or open secondary channels on dynamically assigned ports.

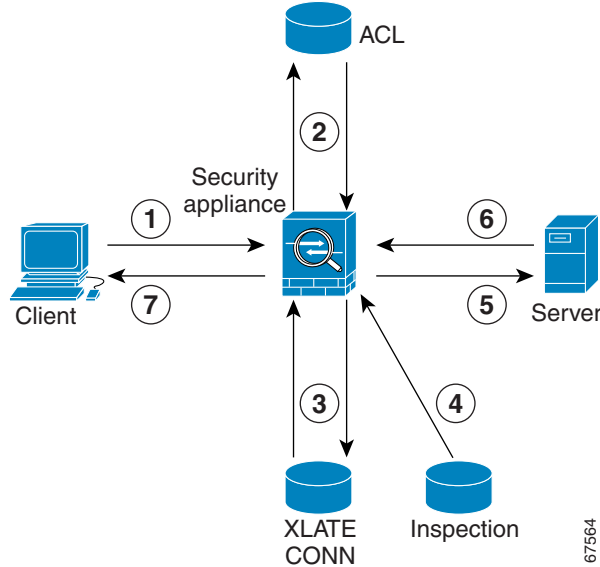
The application inspection function works with NAT to help identify the location of embedded addressing information. This allows NAT to translate these embedded addresses and to update any checksum or other fields that are affected by the translation.

The application inspection function also monitors sessions to determine the port numbers for secondary channels. Many protocols open secondary TCP or UDP ports to improve performance. The initial session on a well-known port is used to negotiate dynamically assigned port numbers. The application inspection function monitors these sessions, identifies the dynamic port assignments, and permits data exchange on these ports for the duration of the specific session.

As illustrated in [Figure 4-1](#), ASA uses three databases for its basic operation:

- Access control lists (ACLs)—Used for authentication and authorization of connections based on specific networks, hosts, and services (TCP/UDP port numbers).
- Inspections—Contains a static, pre-defined set of application-level inspection functions.
- Connections (XLATE and CONN tables)—Maintains state and other information about each established connection. This information is used by ASA and cut-through proxy to efficiently forward traffic within established sessions.

Figure 4-1 Basic ASA Operations



In [Figure 4-1](#), operations are numbered in the order they occur, and are described as follows:

1. A TCP SYN packet arrives at the PIX Firewall to establish a new connection.
2. The PIX Firewall checks the access control list (ACL) database to determine if the connection is permitted.
3. The PIX Firewall creates a new entry in the connection database (XLATE and CONN tables).
4. The PIX Firewall checks the Inspections database to determine if the connection requires application-level inspection.
5. After the application inspection function completes any required operations for the packet, the PIX Firewall forwards the packet to the destination system.
6. The destination system responds to the initial request.
7. The PIX Firewall receives the reply packet, looks up the connection in the connection database, and forwards the packet because it belongs to an established session.

The default configuration of the PIX Firewall includes a set of application inspection entries that associate supported protocols with specific TCP or UDP port numbers and that identify any special handling required. The inspection function does not support NAT or PAT for certain applications because of the constraints imposed by the applications. You can change the port assignments for some applications, while other applications have fixed port assignments that you cannot change. [Table 4-1](#) summarizes this information about the application inspection functions provided with PIX Firewall version 6.2.

Table 4-1 Application Inspection Functions

Application	PAT Support?	NAT (1-1) Support?	Configurable?	Default Port	Related Standards	Limitations/ Comments
H.323	In PIX Firewall version 6.2	Yes	Yes No	TCP/1720 UDP/1718	ITU-T H.323, H.245, H225.0, Q.931, Q.932	None
H.323 RAS	In PIX Firewall version 6.2	Yes	Yes (in version 6.2)	UDP/1719	—	Gatekeeper TCP Control
SIP	In PIX Firewall version 6.2	Yes	Yes No	TCP/5060 UDP/5060	RFC 2543	None
FTP	Yes	Yes	Yes	TCP/21	RFC 1123	None
ILS (LDAP)	Yes	No outside NAT	Yes	—	—	Introduced in PIX Firewall version 6.2
SMTP	Yes	Yes	Yes	TCP/25	RFC 821, 1123	None
SQL*Net	Yes	Yes	Yes	TCP/1521 (v.1)	—	V.1 and v.2
HTTP	Yes	Yes	Yes	TCP/80	RFC 2616	Beware of MTU limitations when stripping ActiveX and Java
RSH	Yes	Yes	Yes	TCP/514	Berkeley UNIX	None
SKINNY (SCCP)	No	Yes	Yes	TCP/2000	—	Does not handle TFTP uploaded configurations
DNS	Yes	Yes	No	UDP/53	RFC 1123	Only forward NAT. No PTR records are changed
NetBIOS over IP	No	No	No	—	—	None
NBNS / UDP	No	No	No	UDP/137	—	No WINS support
NBDS / UDP	Yes	Yes	No	UDP/138	—	None
Sun RPC	No	No	No	UDP/111 TCP/111	—	Payload not NATed
XDCMP	No	No	No	UDP/117	—	None
RTSP	No	No	Yes	TCP/554	RFC 2326, RFC 2327, RFC 1889	No HTTP cloaking handling
CU-SeeMe	No	No	No	UDP/7648	—	None
ICMP	Yes	Yes	No	—	—	None
VDO LIVE	No	Yes	No	TCP/7000	—	None
Windows Media a.k.a. Netshow	No	Yes	No	TCP/1755	—	Can stream over HTTP, TCP or UDP

If the MTU is too small to allow the Java or ActiveX tag to be included in one packet, stripping may not occur.

The PC protocol NetBIOS is supported by performing NAT of the packets for the following services:

- NBNS UDP port 137
- NBDS UDP port 138

No NAT support is available for name resolution through WINS.

Using the fixup Command

You can use the **fixup** command to change the default port assignments or to enable or disable application inspection for the following protocols and applications:

- FTP
- H.323
- HTTP
- ILS
- RSH
- RTSP
- SIP
- SKINNY (SCCP)
- SMTP
- SQL*Net

The basic syntax for the **fixup** command is as follows:

```
[no] fixup protocol [protocol] [port]
```

To change the default port assignment, identify the protocol and the new port number to assign. Use the **no fixup protocol** command to reset the application inspection entries to the default configuration.



Note

Disabling or modifying application inspection only affects connections that are initiated after the command is processed. Disabling application inspection for a specific port or application does not affect existing connections. If you want the change to take effect immediately, enter the **clear xlate** command to remove all existing application inspection entries.

The following is the detailed syntax of the **fixup** command showing the syntax for each configurable application:

```
fixup protocol ftp [strict] [port] | http [port[-port]] | h323 [port[-port]] | ils
[port[-port]] | rsh [514] | rtsp [port] | sip [5060] | skinny [port] | smtp [port[-port]] |
sqlnet [port[-port]]
```

You can view the explicit (configurable) **fixup protocol** settings with the **show fixup** command. The default settings for configurable protocols are as follows.

```
show fixup
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 1720
fixup protocol rsh 514
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
```

The default port value for **rsh** cannot be changed, but additional port statements can be added.

The **show fixup protocol protocol** command displays the configuration for an individual protocol.

The following are other related commands that let you manage fixup configuration:

- **show conn state**—Displays the connection state of the designated protocol
- **show timeout**—Displays the timeout value of the designated protocol

The **clear fixup** command removes **fixup** commands from the configuration that you added. 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.

For some applications, you can define multiple port assignments. This is useful when multiple instances of the same service are running on different ports.

The following example shows how to define multiple ports for FTP by entering separate commands:

```
fixup protocol ftp 2100
fixup protocol ftp 4254
fixup protocol ftp 9090
```

These commands do not change the standard FTP port assignment (21). After entering these commands, the PIX Firewall listens for FTP traffic on port 21, 2100, 4254, and 9090.

Some protocols let you assign a range of ports. This is indicated in the command syntax as port[-port]. For example, the following command assigns the port range from 1500 to 2000 to SQL*Net.

```
fixup protocol sqlnet 1500-2000
```

**Note**

If you enter a new port assignment for protocols that do not allow multiple port assignments, the value overrides the default value.

Basic Internet Protocols

This section describes how the PIX Firewall supports the most common Internet protocols and how you can use the **fixup** command and other commands to solve specific problems. It includes the following topics:

- [File Transfer Protocol](#)
- [Domain Name System](#)
- [Hypertext Transfer Protocol](#)
- [Simple Mail Transfer Protocol](#)

File Transfer Protocol

You can use the **fixup** command to change the default port assignment for the File Transfer Protocol (FTP). The command syntax is as follows:

```
[no] fixup protocol ftp [strict] [port]
```

The **port** parameter lets you configure the port at which the PIX Firewall listens for FTP traffic.

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.

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.

**Note**

The use of the **strict** option may break FTP clients that do not comply with the RFC standards.

The FTP application inspection inspects the FTP sessions and performs four tasks:

- Prepares dynamic secondary data connection
- Tracks **ftp** command-response sequence
- Generates an audit trail
- NATs embedded IP address

FTP application inspection prepares secondary channels for FTP data transfer. The channels are allocated in response to a file upload, a file download, or a directory listing event and must be pre-negotiated. The port is negotiated through the PORT or PASV commands.

If the **strict** option is enabled, each **ftp** command and response sequence is tracked for the following anomalous activity:

- Truncated command—Number of commas in the PORT and PASV reply command is checked to see if it is five. If it is not five, then the PORT command is assumed to be truncated and the TCP connection is closed.
- Incorrect command—Checks the **ftp** command to see if it ends with <CR><LF> characters, as required by the RFC. If it does not, the connection is closed.
- Size of RETR and STOR commands—These are checked against a fixed constant. If the size is greater, then an error message is logged and the connection is closed.

- Command spoofing—The PORT command should always be sent from the client. The TCP connection is denied if a PORT command is sent from the server.
- Reply spoofing—PASV reply command (227) should always be sent from the server. The TCP connection is denied if a PASV reply command is sent from the client. This prevents the security hole when the user executes “227 xxxxx a1, a2, a3, a4, p1, p2.”
- TCP stream editing.
- Invalid port negotiation—The negotiated dynamic port value is checked to see if it is less than 1024. As port numbers in the range from 1 to 1024 are reserved for well known connections, if the negotiated port falls in this range then the TCP connection is freed.
- Command pipelining—The number of characters present after the port numbers in the PORT and PASV reply command is cross checked with a constant value of 8. If it is more than 8, then the TCP connection is closed.

FTP application inspection generates the following log messages:

- An Audit record 302002 is generated for each file that is retrieved or uploaded.
- The **ftp** command is checked to see if it is RETR or STOR and the retrieve and store commands are logged.
- The username is obtained by looking up a table providing the IP address.
- The username, source IP address, destination IP address, NAT address, and the file operation are logged.
- Audit record 201005 is generated if the secondary dynamic channel preparation failed due to memory shortage.

In conjunction with NAT, the FTP application inspection translates the IP address within the application payload. This is described in detail in RFC 959.

Domain Name System

The port assignment for the Domain Name System (DNS) is not configurable. DNS requires application inspection so that DNS queries will not be subject to the generic UDP handling based on activity timeouts. Instead, the UDP connections associated with DNS queries and responses are torn down as soon as a reply to a DNS query has been received. This functionality is called DNS Guard.

DNS inspection performs two tasks:

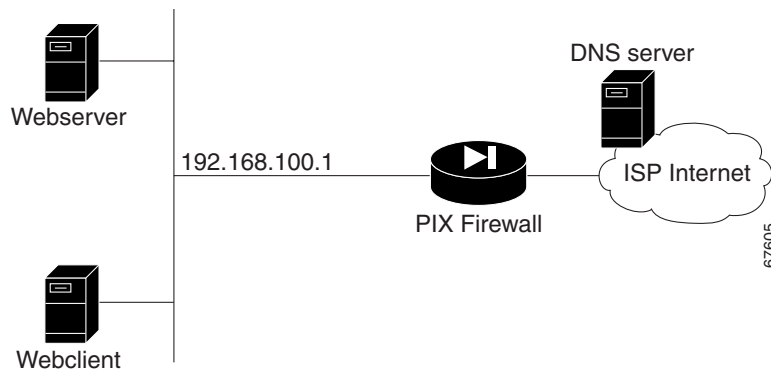
- Monitors the message exchange to ensure that the ID of the DNS reply matches the ID of the DNS query.
- Translates the DNS A-record on behalf of the **alias** command. With PIX Firewall version 6.2, DNS inspection also supports static and dynamic NAT and Outside NAT makes the use of the **alias** command unnecessary.

Only forward lookups are NATed, so PTR records are not touched. Alarms can also be set off in the Intrusion Detection System (IDS) module for DNS zone transfers.

PIX Firewall version 6.2 introduces full support for NAT and PAT of DNS messages originating from either inside (more secure) or outside (less secure) interfaces. This means that if a client on an inside network requests DNS resolution of an inside address from a DNS server on an outside interface, the DNS A-record is translated correctly.

For example, in [Figure 4-2](#), a client on the inside network issues an HTTP request to server 192.168.100.1, using its host name server.example.com. The address of this server is mapped through PAT to a single ISP-assigned address 209.165.200.5. The DNS server resides on the ISP network.

Figure 4-2 NAT/PAT of DNS Messages



When the request is made to the DNS server, the PIX Firewall translates the non-routable source address in the IP header and forwards the request to the ISP network on its outside interface. When the DNS A-record is returned, the PIX Firewall applies address translation not only to the destination address, but also to the embedded IP address of the web server. This address is contained in the user data portion of the DNS reply packet. As a result, the web client on the inside network gets the address it needs to connect to the web server on the inside network.

The transparent support for DNS in PIX Firewall version 6.2 means that the same process works if the client making the DNS request is on a DMZ (or other less secure) network and the DNS server is on an inside (or other more secure) interface.

Hypertext Transfer Protocol

You can use the **fixup** command to change the default port assignment for the Hypertext Transfer Protocol (HTTP). The command syntax is as follows.

```
fixup protocol http [port[-port]]
```

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



Note

The **no fixup protocol http** command statement also disables the **filter url** command.

HTTP inspection performs several functions:

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

The latter two features are described in “[Filtering Outbound Connections](#)” in Chapter 3, “[Controlling Network Access and Use](#).”

Simple Mail Transfer Protocol

This section describes how application inspection works with the Simple Mail Transfer Protocol (SMTP). It includes the following topics:

- [Application Inspection](#)
- [Sample Configuration](#)

You can use the **fixup** command to change the default port assignment for SMTP. The command syntax is as follows.

```
fixup protocol smtp [port[-port]]
```

The **fixup protocol smtp** command enables the Mail Guard feature. This restricts mail servers to receiving the seven minimal commands defined in RFC 821, section 4.5.1 (HELO, MAIL, RCPT, DATA, RSET, NOOP, and QUIT). All other commands are rejected.

Microsoft Exchange server does not strictly comply with RFC 821 section 4.5.1, using extended SMTP commands such as EHLO. PIX Firewall will convert any such commands into NOOP commands, which as specified by the RFC, forces SMTP servers to fall back to using minimal SMTP commands only. This may cause Microsoft Outlook clients and Exchange servers to function unpredictably when their connection passes through PIX Firewall.

Use the *port* option to change the default port assignments from 25. Use the *-port* option to apply SMTP application inspection to a range of port numbers.

As of version 5.1 and higher, the **fixup protocol smtp** command changes the characters in the server SMTP banner to asterisks except for the “2”, “0”, “0” characters. Carriage return (CR) and linefeed (LF) characters are ignored. PIX Firewall version 4.4 converts all characters in the SMTP banner to asterisks.

Application Inspection

An SMTP server responds to client requests with numeric reply codes and optional human readable strings. SMTP application inspection controls and reduces the commands that the user can use as well as the messages that the server returns. SMTP inspection performs three primary tasks:

- Restricts SMTP requests to seven minimal commands (HELO, MAIL, RCPT, DATA, RSET, NOOP, and QUIT).
- Monitors the SMTP command-response sequence.
- Generates an audit trail—Audit record 108002 is generated when invalid character embedded in the mail address is replaced. For more information, see RFC 821.

SMTP inspection monitors the command and response sequence for the following anomalous signatures:

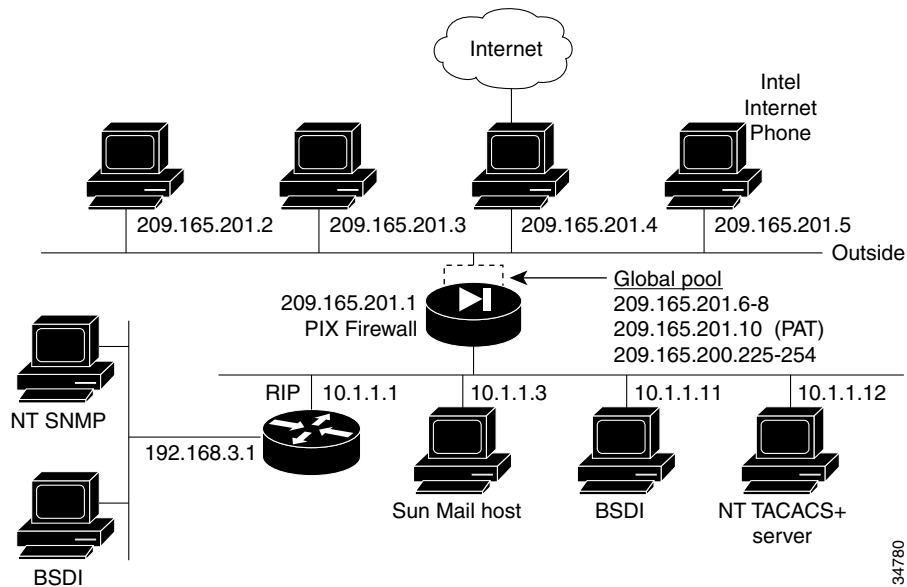
- Truncated commands.
- Incorrect command termination (not terminated with <CR><LR>).
- The MAIL and RCPT commands specify who are the sender and the receiver of the mail. Mail addresses are scanned for strange characters. The pipeline character (|) is deleted (changed to a blank space) and “<” ,”>” are only allowed if they are used to define a mail address (“>” must be preceded by “<”).
- Unexpected transition by the SMTP server.

- For unknown commands, the PIX Firewall changes all the characters in the packet to X. In this case, the server will generate an error code to the client. Because of the change in the packed, the TCP checksum has to be recalculated or adjusted.
- TCP stream editing.
- Command pipelining.

Sample Configuration

Figure 4-3 illustrates a network scenario implementing SMTP and NFS on an internal network.

Figure 4-3 Sample Configuration with SMTP and NFS (Sun RPC)



In this example, the **static** command sets up a global address to permit outside hosts access to the 10.1.1.3 Sun Mail host on the Inside interface. (The MX record for DNS must 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.

Perform the following steps to complete the configuration required for this example:

Step 1 Provide access to the 10.1.1.3 mail server through global address 209.165.201.12:

```
static (inside, outside) 209.165.201.12 10.1.1.3 netmask 255.255.255.255 0 0
access-list acl_out permit tcp any host 209.165.201.12 eq smtp
```

The **access-list** command allows any outside host access to the static via SMTP (port 25). By default, the PIX Firewall restricts all access to mail servers to the commands DATA, HELO, MAIL, NOOP, QUIT, RCPT, and RSET, as described in RFC 821, section 4.5.1. This is implemented through the Mail Guard service, which is enabled by default (**fixup protocol smtp 25**).

Another aspect of providing access to a mail server is being sure that you have a DNS MX record for the static's global address, which outside users access when sending mail to your site.

Step 2 Create access to port 113, the IDENT protocol:

```
access-list acl_out permit tcp any host 209.165.201.12 eq 113
access-group acl_out in interface outside
static (inside, outside) 209.165.201.12 10.1.1.3 netmask 255.255.255.0 0
access-list acl_out permit tcp any host 209.165.201.12 eq smtp
access-list acl_out permit tcp any host 209.165.201.12 eq 113
access-group acl_out in interface outside
```

If the mail server has to talk to many mail servers on the outside which connect back with the now obsolete and highly criticized IDENT protocol, use this **access-list** command statement to speed up mail transmission. The **access-group** command statement binds the **access-list** command statements to the outside interface.

[Example 4-1](#) shows a command listing for configuring access to services for the network:

Example 4-1 Configuring Mail Server Access

```
static (inside, outside) 209.165.201.12 10.1.1.3 netmask 255.255.255.0 0
access-list acl_out permit tcp any host 209.165.201.12 eq smtp
access-list acl_out permit tcp any host 209.165.201.12 eq 113
access-group acl_out in interface outside
static (inside, outside) 209.165.201.12 10.1.1.3 netmask 255.255.255.0 0
```

Voice Over IP

This section describes how the PIX Firewall supports Voice over IP (VoIP) applications and protocols and how you can use **fixup** and other commands to solve specific problems. It includes the following topics:

- [Skinny Client Control Protocol](#)
- [H.323](#)
- [Session Initiation Protocol](#)
- [CU-SeeMe](#)

Skinny Client Control Protocol

Skinny (or Simple) Client Control Protocol (SCCP) is a simplified protocol used in VoIP networks. This section describes the function and limitation of application inspection when using SCCP. It includes the following topics:

- [Overview](#)
- [Using SCCP with Cisco CallManager on a Higher Security Interface](#)
- [Problems Occur with Fragmented SCCP Packets](#)

Overview

Cisco IP Phones using SCCP can coexist in an H.323 environment. When used with Cisco CallManager, the SCCP client can interoperate with H.323 compliant terminals. Application layer functions in the PIX Firewall recognize SCCP version 3.1.1. The functionality of the application layer software ensures that all SCCP signalling and media packets can traverse the Firewall by providing NAT of the SCCP Signaling packets.

You can use the **fixup** command to change the default port assignment for SCCP. The command syntax is as follows.

```
[no] fixup protocol skinny [port[-port]]
```

To change the default port assignments from 2000 use the *port* option. Use the *-port* option to apply SCCP application inspection to a range of port numbers.



Note

If the address of a Cisco CallManager server is configured for NAT and outside phones register to it using TFTP, the connection will fail because PIX Firewall currently does not support NAT TFTP messages. For a workaround to this problem, refer to the subsection “[Using SCCP with Cisco CallManager on a Higher Security Interface](#)” within this section.

The IP addresses need to be configured for allowable outside interfaces that can initiate calls or receive RTP packets. SCCP is not supported through PAT, but is supported with NAT.

PIX Firewall version 6.2 introduces support of DHCP options 150 and 166, which allow the PIX Firewall to send the location of a TFTP server to Cisco IP Phones and other DHCP clients. For further information about this new feature, refer to “[Using Cisco IP Phones with a DHCP Server](#)” in [Chapter 5, “Using PIX Firewall in SOHO Networks.”](#)

Using SCCP with Cisco CallManager on a Higher Security Interface

The PIX Firewall does not support TFTP application inspection, so NAT and PAT cannot be used to translate the address of a TFTP server on an inside or higher security interface. Cisco IP Phones require access to a TFTP server to download the configuration information they need to connect to the Cisco CallManager server. Typically, this TFTP service runs on the same machine as Cisco CallManager.

Cisco CallManager is often implemented at a central site to control Cisco IP Phones distributed at branch offices. In this scenario, the Cisco IP Phones at the branch offices need TFTP access through the interface to which the Cisco CallManager server communicates. You can provide this access in one of the following ways:

- Create an access list that allows connections to be initiated on the TFTP port (UDP 69) from each branch network subnet
- Create a static entry without NAT to allow access to the IP address of the TFTP server on the outside interface



Note

Normal traffic between the Cisco CallManager and Cisco IP Phones uses SCCP and is handled by SCCP inspection without any special configuration.

Problems Occur with Fragmented SCCP Packets

At this time, PIX Firewall is not able to correctly handle fragmented SCCP packets. For instance, when using a voice conferencing bridge, SCCP packets may become fragmented and are then dropped by the PIX Firewall. This happens because the SCCP inspection checks each packet and drops what appear to be bad packets. When a single SCCP packet is fragmented into multiple TCP packets, the SCCP inspection function finds that the internal checksums within the SCCP packet fragments are not accurate and so it drops the packet.

H.323

You can use the **fixup** command to change the default port assignment for the H.323 protocol. The command syntax is as follows:

```
[no] fixup protocol h323 h225 [ras port [-port]]
```

Use the *port* option to change the default control connection port assignment. The default port assignments are as follows:

- h323 h225 1720
- h323 ras 1718-1719

Use the *-port* option to apply H.323 application inspection to a range of port numbers.

The **fixup protocol h323** command provides support for Intel Internet Phone, CU-SeeMe, CU-SeeMe Pro, MeetingPoint, and MS NetMeeting. PIX Firewall version 5.3 and higher supports H.323 version 2. H.323 is a suite of protocols defined by the International Telecommunication Union (ITU) for multimedia conferences over LANs. H.323 supports VoIP gateways and VoIP gatekeepers. H.323 version 2 adds the following functionality:

- Fast Connect or Fast Start Procedure for faster call setup
- H.245 tunneling for resource conservation, call synchronization, and reduced set up time

H.323 inspection supports static NAT or dynamic NAT. H.323 RAS is configurable using the **fixup** command with PIX Firewall version 6.2 or later. With earlier versions, only H.225 & H.245 signaling can be controlled using the **fixup** command. PAT support for H.323 is introduced with PIX Firewall version 6.2.

The H.323 collection of protocols collectively may use up to two TCP connection and four to six UDP connections. FastConnect uses only one TCP connection, and RAS uses a single UDP connection for registration, admissions, and status.

An H.323 client may initially establish a TCP connection to an H.323 server using TCP port 1720 to request Q.931 call setup. As part of the call setup process, the H.323 terminal supplies a port number to the client to use for an H.245 TCP connection. In environments where an H.323 gatekeeper is in use, the initial packet is transmitted using UDP, where the client sends out an ARQ.

H.323 inspection monitors the Q.931 TCP connection to determine the H.245 port number. If the H.323 terminals are not using FastConnect, the PIX Firewall dynamically allocates the H.245 connection based on the inspection of the H.225 messages.

Within each H.245 message, the H.323 end points exchange port numbers that are used for subsequent UDP data streams. H.323 inspection inspects the H.245 messages to identify these ports and dynamically creates connections for the media exchange. Real-Time Transport Protocol (RTP) uses the negotiated port number, while RTP Control Protocol (RTCP) uses the next higher port number.

The H.323 control channel handles H.225 and H.245 and H.323 RAS. H.323 inspection uses the following ports.

- 1718—Gate Keeper Discovery UDP port
- 1719—RAS UDP port
- 1720—TCP Control Port

The two major functions of H.323 inspection are as follows:

- NAT the necessary embedded IPv4 addresses in the H.225 and H.245 messages. Because H.323 messages are encoded in PER encoding format, PIX Firewall uses an ASN.1 decoder to decode the H.323 messages.
- Dynamically allocate the negotiated H.245 and RTP/RTCP connections.

The PIX Firewall administrator must open a conduit for the well-known H.323 port 1720 for the H.225 call signaling. However, the H.245 signaling ports are negotiated between the endpoints in the H.225 signaling. When an H.323 gatekeeper is used, the PIX Firewall opens an H.225 connection based on inspection of the ACF message.

The PIX Firewall dynamically allocates the H.245 channel after inspecting the H.225 messages and then “hookup” the H.245 channel to be fixed up as well. That means whatever H.245 messages pass through the PIX Firewall pass through the H.245 application inspection, NATing embedded IP addresses and opening the negotiated media channels.

The H.323 ITU standard requires that a TPKT header, defining the length of the message, precede the H.225 and H.245, before being passed on to the reliable connection. Because the TPKT header does not necessarily need to be sent in the same TCP packet as the H.225/H.245 message, PIX Firewall must remember the TPKT length to process/decode the messages properly. PIX Firewall keeps a data structure for each connection and that data structure contains the TPKT length for the next expected message.

If the PIX Firewall needs to NAT any IP addresses, then it will have to change the checksum, the UUIE (user-user information element) length, and the TPKT, if included in the TCP packet with the H.225 message. If the TPKT is sent in a separate TCP packet, then PIX Firewall will proxy ACK that TPKT and append a new TPKT to the H.245 message with the new length.



Note

PIX Firewall does not support TCP options in the Proxy ACK for the TPKT.

Each UDP connection with a packet going through H.323 inspection is marked as an H.323 connection and will time out with the H.323 timeout as configured by the administrator using the **timeout** command.

Usage Notes

1. Static PAT may not properly translate IP addresses embedded in optional fields within H.323 messages. If you experience this kind of problem, do not use static PAT with H.323.
2. When a NetMeeting client registers with an H.323 gatekeeper and tries to call an H.323 gateway that is also registered with the H.323 gatekeeper, the connection is established but no voice is heard in either direction. This problem is unrelated to the PIX Firewall.
3. If you configure a network static where the network static is the same as a third-party netmask and address, then any outbound H.323 connection fails.

Session Initiation Protocol

Session Initiation Protocol (SIP), as defined by the Internet Engineering Task Force (IETF), enables call handling sessions, particularly two-party audio conferences, or “calls.” This section describes how application inspection works with SIP. It includes the following topics:

- [Overview](#)
- [Allowing Outside Phones to Place an Inside Phone on Hold](#)
- [Technical Details](#)

Overview

SIP works with Session Description Protocol (SDP) for call signalling. SDP specifies the ports for the media stream. Using SIP, the PIX Firewall can support any SIP Voice over IP (VoIP) gateways and VoIP proxy servers. SIP and SDP are defined in the following RFCs:

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

You can use the **fixup** command to change the default TCP port assignment for the Session Initiation Protocol (SIP). The command syntax is as follows.

```
[no] fixup protocol sip [port[-port]]
```



Note

PAT support for SIP is introduced with PIX Firewall version 6.2.

To change the default port assignments from 5060 use the *port* option. Use the *-port* option to apply SIP application inspection to a range of port numbers.

To view the current timeout value for SIP connections, enter the following command:

```
show timeout sip
```

To view the state of SIP connections, enter the following command:

```
show conn state sip
```



Note

If a remote endpoint tries to register with a SIP proxy on a network protected by PIX Firewall, the registration will fail if the To field in the request does not specify the port number and if the SIP proxy is configured with PAT.

To support SIP calls through the PIX Firewall, signaling messages for the media connection addresses, media ports, and embryonic connections for the media must be inspected, because while the signaling is sent over a well known destination port (UDP/TCP 5060), the media streams are dynamically allocated. Also, SIP embeds IP addresses in the user-data portion of the IP packet. SIP inspection applies NAT for these embedded IP addresses.

With SIP application inspection enabled, the PIX Firewall does support connectivity between a SIP phone and a Music on Hold (MOH) server. The specific scenario that has been tested is with a phone on the more secure network connected to an MOH server with the SIP proxy on the less secure network.



Note

Application inspection of UDP for SIP is always enabled—it is currently not configurable.

Allowing Outside Phones to Place an Inside Phone on Hold

When an outbound call is made by an IP Phone using SIP and the outside phone tries to put the inside phone on hold, the operation fails. This is because a new connection is initiated to send the INVITE packet from the outside phone and the PIX Firewall drops the packet.

To solve this problem, do one of the following:

- Configure an access list to allow the Re-INVITE packet to the inside gateway using port 5060
- Use the **established** command, as in the following example:

```
established dup 5060 permitto udp 5060 permitfrom udp 0
```

This command statement causes the PIX Firewall to allow a new connection on port 5060 from an outside phone if a UDP connection already exists from that phone to an inside phone. A call can be placed on hold for the time specified in the timeout interval for SIP. You can increase this interval as necessary with the **timeout** command.

Technical Details

SIP inspection NATs the SIP text-based messages, recalculates the content length for the SDP portion of the message, and recalculates the packet length and checksum. It dynamically opens media connections for ports specified in the SDP portion of the SIP message as address/ports on which the endpoint should listen.

SIP inspection has a database with indices CALL_ID/FROM/TO from the SIP payload that identifies the call. Contained within this database are the media addresses and media ports that were contained in the SDP media information fields and the media type. There can be multiple media addresses and ports for a session. RTP/RTCP connections are opened between the two endpoints using these media addresses/ports. The well-known port 5060 must be used on the initial call setup (INVITE) message, however subsequent messages may not have this port number. The SIP fixup opens signaling connection pinholes, and marks these connections as SIP connections. This is done for the messages to reach the SIP application and be NATed.

As a call is set up, the SIP session is considered in the “transient” state until the media address and media port is received in a Response message from the called endpoint indicating the RTP port the called endpoint will listen on. If there is a failure to receive the response messages within one minute, the signaling connection will be torn down.

Once the final handshake is made, the call state is moved to active and the signaling connection will remain until a BYE message is received.

If an inside endpoint initiates a call to an outside endpoint, a media hole is opened to the outside interface to allow RTP/RTCP UDP packets to flow to the inside endpoint media address and media port specified in the INVITE message from the inside endpoint. Unsolicited RTP/RTCP UDP packets to an inside interface will not traverse the Firewall, unless the PIX Firewall Configuration specifically allows it.

The media connections are torn down within two minutes after the connection becomes idle. This is, however, a configurable timeout and can be set for a shorter or longer period of time.



Note

Support for PAT is introduced with PIX Firewall version 6.2. Static NAT and dynamic NAT are supported in version 6.2 and earlier.

CU-SeeMe

With CU-SeeMe clients, one user can connect directly to another (CU-SeeMe or other H.323 client) for person-to-person audio, video, and data collaboration. CU-SeeMe clients can conference in a mixed client environment that includes both CU-SeeMe clients and H.323-compliant clients from other vendors.

Behind the scenes, CU-SeeMe clients operate in two very different modes. When connected to another CU-SeeMe client or CU-SeeMe Conference Server, the client sends information in CU-SeeMe mode.

When connected to an H.323-compliant videoconferencing client from a different vendor, CU-SeeMe clients communicate using the H.323-standard format in H.323 mode.

CU-SeeMe is supported through H.323 inspection, as well as performing NAT on the CU-SeeMe control stream, which operates on UDP port 7648.

Multimedia Applications

This section describes how the PIX Firewall supports multimedia or video-on-demand applications and protocols and how you can use **fixup** and other commands to solve specific problems. It includes the following topics:

- [Real Time Streaming Protocol \(RTSP\)](#)
- [Netshow](#)
- [VDO LIVE](#)

Real Time Streaming Protocol (RTSP)

You can use the **fixup** command to change the default port assignment for the Real Time Streaming Protocol (RTSP). The command syntax is as follows.

```
fixup rtsp [port]
```

The **fixup protocol rtsp** command lets PIX Firewall pass RTSP packets. RTSP is used by RealAudio, RealNetworks, Apple QuickTime 4, RealPlayer, and Cisco IP/TV connections. PIX Firewall does not support multicast RTSP.

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:

- This PIX Firewall will not fix RTSP messages passing through UDP ports.
- PIX Firewall does not support RealNetworks multicast mode (x-real-rdt/mcast).
- PAT is not supported with the **fixup protocol rtsp** command.
- PIX Firewall does not have the ability to recognize HTTP cloaking where RTSP messages are hidden in the HTTP messages.
- 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.

- 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).
- 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.
- 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. On the PIX Firewall, add a **fixup protocol rtsp port** command statement.

RTSP applications use the well-known port 554 with TCP (rarely UDP) as a control channel. PIX Firewall only supports TCP, in conformity with RFC 2326.

This TCP control channel will be used to negotiate the data channels that will be used to transmit audio/video traffic, depending on the transport mode that is configured on the client.

The supported RDT transports are: rtp/avp, rtp/avp/udp, x-real-rdt, x-real-rdt/udp, x-pn-tng/udp

The PIX Firewall parses Setup response messages with a status code of 200. If the response message is travelling inbound, the server is outside relative to the PIX Firewall and dynamic channels need to be opened for connections coming inbound from the server. If the response message is outbound, then the PIX Firewall does not need to open dynamic channels.

Because RFC 2326 does not require that the client and server ports must be in the SETUP response message, the PIX Firewall will need to keep state and remember the client ports in the SETUP message. QuickTime places the client ports in the SETUP message and then the server responds with only the server ports.

RTSP inspection does not support PAT or dual-NAT. Also, PIX Firewall cannot recognize HTTP cloaking where RTSP messages are hidden in the HTTP messages.

Netshow

Netshow is a streaming multimedia service that allows users to receive audio and video streams from across the Internet. Users play Netshow content using Windows Media player, which connects to the Netshow server to receive the multimedia stream.

The data channel in which the streams are transmitted is negotiated in a control channel. There are different protocol settings possible.

- UDP Stream
- TCP Stream

UDP Stream

UDP streams are used with Netshow as follows:

1. Client makes a TCP connection to the server at the well-known port 1755.
2. Once a connection is established, the client sends an LVMConnectFunnel message to the server indicating the UDP port that it expects to receive the data.
3. Server chooses a UDP port in the range 1024-5000 to stream the netshow data down to the client.
4. Server sends the stream in the negotiated port.
5. Netshow session ends by tearing down the TCP connection.

TCP Stream

TCP streams are used with Netshow as follows:

1. Client makes a TCP connection to the server using the well-known port 1755.
2. Once a connection is established, the client sends an LVMConnectFunnel message to the server confirming the use of TCP connection.
3. Server sends the stream in the already connected TCP port.
4. Netshow session ends by tearing down the TCP connection.

VDO LIVE

VDO LIVE is a streaming multimedia service that allows users to receive audio and video streams from across the Internet.

There are two connections, TCP for control messages and UDP for streams. TCP session uses a fixed port of 7000; while the UDP source port is always 7001. The UDP stream uses a destination port provided by the client over the control connection.

PIX Firewall monitors the VDO Live TCP control session and allows only the VDO live server system to communicate with the client via the solicited UDP port with source port 7001. During this time, the TCP channel should be active. When one goes down, PIX Firewall tears down the other connection.

PIX Firewall bypasses the data channel by opening up the port that was negotiated in the control channel. The application inspection scans the control channel and opens up the negotiated ports.

When NAT is involved, the negotiated IP address and the port number is NAT translated, which means that the payload has to be modified.

Database and Directory Support

This section describes how to allow access to database or directory services through the PIX Firewall. It includes the following topics:

- [Network File System and Sun RPC](#)
- [Oracle SQL*Net \(V1/V2\)](#)
- [ILS and LDAP](#)

Network File System and Sun RPC

The port assignment for Sun Remote Procedure Call (RPC) is not configurable. Sun RPC is used by Network File System (NFS) and Network Information Service (NIS).

Sun RPC services can run on any port on the system. When a client attempts to access a RPC service on a server, it must find out which port that service is running on. It does this by querying the portmapper process on the well-known port of 111.

The client sends the RPC program number of the service, and gets back the port number. From this point on, the client program will send its RPC queries to that new port.

Only frames going from inside to outside are inspected. (for example, the portmapper service running on one of the internal servers has sent a reply). When a server behind the firewall (on the inside interface) sends out a reply, PIX Firewall intercepts this packet and opens both embryonic TCP and UDP connections on that port.

NAT or PAT of RPC payload information is not supported.

**Note**

The `sunrpc` fixup only inspects the original portmapper connection if it is over UDP. TCP portmapper traffic is not inspected.

The following commands demonstrate how to implement Network File System (NFS) for the network shown in [Figure 4-3](#). These commands are used in addition to the basic firewall configuration required:

- Step 1** Refine the accessibility of the `static` command by permitting Sun RPC over the UDP portmapper on port 111 with the `sunrpc` literal:

```
access-list acl_out permit udp host 209.165.201.2 host 209.165.201.11 eq sunrpc
```

Refer to the UNIX `/etc/rpc` file and the UNIX `rpc(3N)` command page for more information.

Once you create an `access-list` command statement for RPC, you can use the following command from outside host 209.165.201.2 to track down the activity of a PCNFSD on RPC 150001:

```
rpcinfo -u 209.165.201.11 150001
```

Another use of RPC is with the following command to see the exports of 209.165.201.11 if you want to allow mounting NFS from the outside network to the inside network:

```
showmount -e 209.165.201.11
```

Many protocols based on RPC, as well as NFS, are insecure and should be used with caution. Review your security policies carefully before permitting access to RPC.

- Step 2** Permit NFS access:

```
access-list acl_out permit udp host 209.165.201.2 host 209.165.201.11 eq 2049
```

NFS access occurs at port 2049 and provides access between the outside and inside, such that host 209.165.201.2 can mount 10.1.1.11 via the global address 209.165.201.11.

[Example 4-2](#) shows the command listing for configuring access to services for the network illustrated in [Figure 4-3](#).

Example 4-2 Configuring NFS Access

```
access-list acl_out permit udp host 209.165.201.2 host 209.165.201.11 eq sunrpc
access-list acl_out permit udp host 209.165.201.2 host 209.165.201.11 eq 2049
```

Oracle SQL*Net (V1/V2)

You can use the **fixup** command to change the default port assignment for Oracle SQL*Net. The command syntax is as follows.

```
fixup protocol sqlnet [port[-port]]
```

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.

The PIX Firewall NATs all addresses and looks in the packets for all embedded ports to open for SQL*Net Version 1.

For SQL*Net Version 2, all DATA or REDIRECT packets that immediately follow REDIRECT packets with a zero data length will be fixed up.

The packets that need fix-up contain embedded host/port addresses in the following format:

```
(ADDRESS=(PROTOCOL=tcp) (DEV=6) (HOST=a.b.c.d) (PORT=a) )
```

SQL*Net Version 2 TNSFrame types (Connect, Accept, Refuse, Resend, and Marker) will not be scanned for addresses to NAT nor will inspection open dynamic connections for any embedded ports in the packet.

SQL*Net Version 2 TNSFrames, Redirect, and Data packets will be scanned for ports to open and addresses to NAT, if preceded by a REDIRECT TNSFrame type with a zero data length for the payload. When the Redirect message with data length zero passes through the PIX Firewall, a flag will be set in the connection data Structure to expect the Data or Redirect message that follows to be NATed and ports to be dynamically opened. If one of the TNS frames in the preceding paragraph arrive after the Redirect message, the flag will be reset.

The SQL*Net fixup will recalculate the checksum, change IP, TCP lengths, and readjust Sequence Numbers and Acknowledgment Numbers using the delta of the length of the new and old message.

SQL*Net Version 1 is assumed for all other cases. TNSFrame types (Connect, Accept, Refuse, Resend, Marker, Redirect, and Data) and all packets will be scanned for ports and addresses. Addresses will be NATed and port connections will be opened.

ILS and LDAP

The Internet Locator Service (ILS) is based on the Lightweight Directory Access Protocol (LDAP) and is LDAPv2 compliant. ILS was developed by Microsoft for use with its NetMeeting, SiteServer, and Active Directory products.

You can use the **fixup** command to change the default port assignment for ILS. The command syntax is as follows.

```
[no] fixup protocol ils [port[-port]]
```

Use the *port* option to change the default port assignment from 389. Use the *-port* option to apply ILS inspection to a range of port numbers.

To show the configuration of ILS inspection, enter the following command:

```
show fixup [protocol ils]
```

PIX Firewall version 6.2 introduces NAT support for ILS, which is used to register and locate endpoints in the ILS or SiteServer Directory. PAT cannot be supported because only IP addresses are stored by an LDAP database.

For Search Responses, when the LDAP server is located outside, NAT should be considered to allow internal peers to communicate locally while registered to external LDAP servers. For such search responses, xlates will be searched first, and then DNAT entries to obtain the correct address. If both of these searches fail, then the address will not be changed. For sites using NAT 0 (no NAT) and not expecting DNAT interaction, we recommend that the fixup be turned off to provide better performance.

Additional configuration may be necessary when the ILS server is located inside the PIX Firewall border. This would require a hole for outside clients to access the LDAP server on the specified port, typically TCP 389.

ILS/LDAP follows a client/server model with sessions handled over a single TCP connection. Depending on the client's actions, several of these sessions may be created.

During connection negotiation time, a BIND PDU is sent from the client to the server. Once a successful BIND RESPONSE from the server is received, other operational messages may be exchanged (such as ADD, DEL, SEARCH, or MODIFY) to perform operations on the ILS Directory. The ADD REQUEST and SEARCH RESPONSE PDUs may contain IP addresses of NetMeeting peers, used by H.323 (SETUP and CONNECT messages) to establish the NetMeeting sessions. Microsoft NetMeeting v2.X and v3.X provides ILS support.

The ILS inspection performs the following operations:

- Decodes the LDAP REQUEST/RESPONSE PDUs using the BER decode functions
- Parses the LDAP packet
- Extracts IP addresses
- Translates IP addresses as necessary
- Encodes the PDU with translated addresses using BER encode functions
- Copies the newly encoded PDU back to the TCP packet
- Performs incremental TCP checksum and sequence number adjustment

ILS inspection has the following limitations:

- Referral requests and responses are not supported
- Users in multiple directories are not unified
- Single users having multiple identities in multiple directories cannot be recognized by NAT

Management Protocols

This section describes how the PIX Firewall supports management protocols to solve specific problems. It includes the following topics:

- [Internet Control Message Protocol](#)
- [Remote Shell](#)
- [X Display Manager Control Protocol](#)

Internet Control Message Protocol

The ICMP payload is scanned to retrieve the five-tuple from the original packet. ICMP inspection supports both 1-1 NAT and PAT. Using the retrieved five tuple, a lookup is performed to determine the original address of the client. ICMP inspection makes the following changes to the ICMP packet:

- In the IP Header, the NAT IP is changed to Client IP (Destination Address) and the IP checksum is modified.
- In ICMP Header, the ICMP checksum is modified due to the changes in the ICMP packet.
- In the Payload the following changes are made:
 - Original packet NAT IP is changed to Client IP
 - Original packet NAT port is changed to Client Port
 - Original packet IP checksum

Remote Shell

You can use the **fixup** command to change the default port assignment for the Remote Shell protocol (RSH). The command syntax is as follows.

```
fixup protocol rsh [514]
```

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.

X Display Manager Control Protocol

The port assignment for the X Display Manager Control Protocol (XDMCP) is not configurable. XDMCP is a protocol that uses UDP port 177 to negotiate X sessions, which use TCP when established.

For successful negotiation and start of an Xwindows session, the PIX Firewall must allow the TCP back connection from the Xhosted computer. To permit the back connection use the **established** command on the PIX Firewall. Once XDMCP negotiates the port to send the display, The **established** command is consulted to verify if this back connection should be permitted.

During the X Windows session, the manager talks to the display's Xserver on the well-known port 6000 + n. Each display has a separate connection to the Xserver, as a result of the following terminal setting.

```
setenv DISPLAY Xserver:n
```

where *n* is the display number.

When XDMCP is used, the display is negotiated using IP addresses, which the PIX Firewall can NAT if needed. XDMCP inspection does not support PAT.

