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

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.

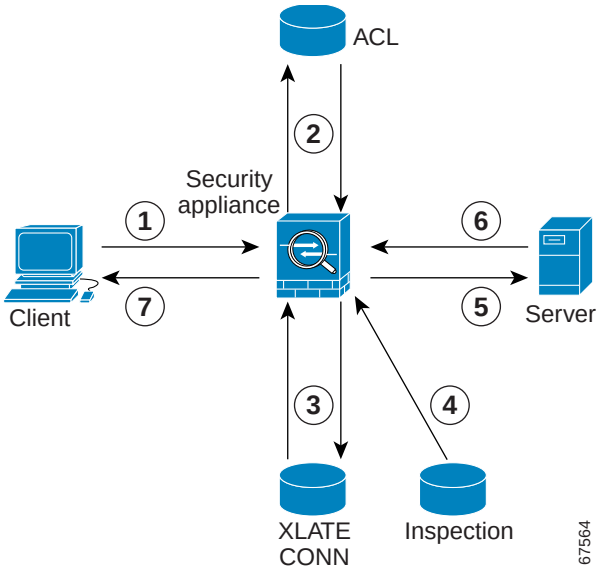
The application inspection function works with NAT to help identify the location of embedded addressing information. This allows NAT to translate these embedded addresses and to update any checksum or other fields that are affected by the translation.

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 5-1, 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 5-1 Basic ASA Operations



In Figure 5-1, 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 5-1 summarizes this information about the application inspection functions provided with PIX Firewall Version 6.2 and higher.

Table 5-1 Application Inspection Functions

Application	PAT?	NAT (1-1)?	Configure?	Default Port	Standards	Limitations/Comments
CTIQBE	Yes	Yes	Yes	TCP/2748	—	Introduced with PIX Firewall Version 6.3
CU-SeeMe	No	No	No	UDP/7648	—	None.
DNS[1]	Yes	Yes	No	UDP/53	RFC 1123	Only forward NAT. No PTR records are changed.

Table 5-1 Application Inspection Functions (continued)

Application	PAT?	NAT (1-1)?	Configure?	Default Port	Standards	Limitations/Comments
FTP	Yes	Yes	Yes	TCP/21	RFC 1123	None.
H.323	PIX Firewall Version 6.2 and higher	Yes	Yes	TCP/1720 UDP/1718 UDP (RAS) 1718-1719	ITU-T H.323, H.245, H225.0, Q.931, Q.932	None. Support for Version 3 and 4 introduced with PIX Firewall Version 6.3. Does not support segmented messages.
HTTP	Yes	Yes	Yes	TCP/80	RFC 2616	Beware of MTU limitations when stripping ActiveX and Java.[2]
ICMP	Yes	Yes	No	—	—	None.
ILS (LDAP)	Yes	Yes	Yes	—	—	Introduced in PIX Firewall Version 6.2.
MGCP	No	No	Yes	2427, 2727	RFC2705bis-05	Introduced with PIX Firewall Version 6.3.
NBDS / UDP	Yes	Yes	No	UDP/138	—	None.
NBNS / UDP	No	No	No	UDP/137	—	No WINS support.
NetBIOS over IP[3]	No	No	No	—	—	None.
PPTP	Yes	Yes	Yes	1723	RFC2637	Introduced with PIX Firewall Version 6.3.
RSH	Yes	Yes	Yes	TCP/514	Berkeley UNIX	None.
RTSP	No	No	Yes	TCP/554	RFC 2326, RFC 2327, RFC 1889	No handling for HTTP cloaking.
SIP	PIX Firewall Version 6.2 or higher	Yes	Yes	TCP/5060 UDP/5060	RFC 2543	None.
SKINNY (SCCP)	PIX Firewall Version 6.3	Yes	Yes	TCP/2000	—	Does not handle TFTP uploaded Cisco IP Phone configurations under certain circumstances.
SMTP	Yes	Yes	Yes	TCP/25	RFC 821, 1123	None.
SNMP	No	No	Yes	UDP 161, 162	RFC 1155, 1157, 1212, 1213, 1215	v.2 RFC 1902-1908; v.3 RFC 2570-2580.
SQL*Net	Yes	Yes	Yes	TCP/1521 (v.1)	—	V.1 and v.2.
Sun RPC	No	No	No	UDP/111 TCP/111	—	Payload not NATed.
VDO LIVE	No	Yes	No	TCP/7000	—	None.
Windows Media	No	Yes	No	TCP/1755	—	Can stream Netshow over HTTP, TCP or UDP.
XDCMP	No	No	No	UDP/117	—	None.

1. No NAT support is available for name resolution through WINS.
2. If the MTU is too small to allow the Java or ActiveX tag to be included in one packet, stripping may not occur.
3. NetBIOS is supported by performing NAT of the packets for NBNS UDP port 137 and NBDS UDP port 138.

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:

- CTIQBE (disabled by default)
- DNS
- ESP-IKE (disabled by default)
- FTP
- H.323
- HTTP
- ILS
- MGCP (disabled by default)
- PPTP (disabled by default)
- RSH
- RTSP
- SIP
- SKINNY (SCCP)
- SMTP
- SNMP
- SQL*Net
- TFTP

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. If there are no **xlates**, such as **nat 0 access-list**, use **clear local-host** instead of **clear xlate** to disable or modify application inspection.

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

fixup protocol ctiqbe 2748 | dns [maximum-length *max-len***] | esp-ike | ftp** [**strict**] [*port*] | **http [***port***[-***port***]] | h323 h225 | ras [***port***[-***port***]] | ils [***port***[-***port***]] | mgcp** [*port*[*-port*]**| pptp 1723 | rsh** [**514**] | **rtsp** [*port*] | sip udp [port] | **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.

```
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)#
```

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 connections with the 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 first command example assigns the port range from 1500 to 2000 to SQL*Net. The second command example shows a smaller port range 161 to 162 for SNMP.

```
fixup protocol sqlnet 1500-2000
fixup protocol snmp 161-162
```

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:

DNS

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 the following 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 and higher, DNS inspection also supports static and dynamic NAT and outside NAT makes the use of the **alias** command unnecessary.
- Reassembles the DNS packet to verify its length. Since DNS packets up to 65535 bytes are permitted to traverse the PIX Firewall, reassembly is done to verify that the packet length is less than the maximum length specified by the user. Otherwise, the packet is dropped.

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.

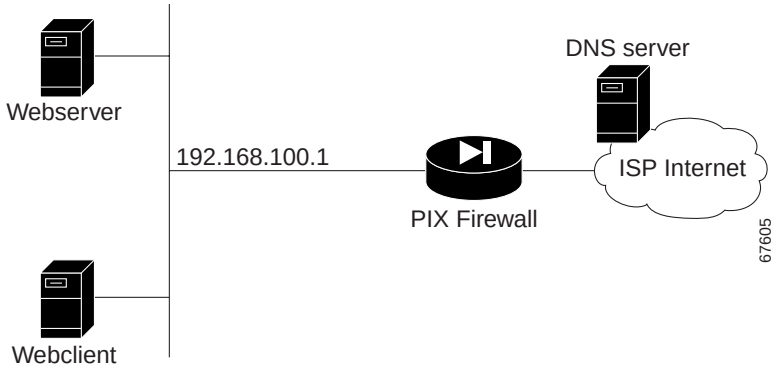
Note The PIX Firewall drops DNS packets sent to UDP port 53 that are larger than the configured maximum length. The default value is 512 bytes.

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. However, fragmented DNS packets will not go though since reassembling is done only if the fixup is turned on.

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 5-2, 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 5-2 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 and higher 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.

FTP

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.

HTTP

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."

ICMP

PIX Firewall Version 6.3 introduces support for NAT of ICMP error messages. NAT for ICMP is disabled by default. When this feature is enabled, the PIX Firewall creates xlates for intermediate hops that send ICMP error messages, based on the static/NAT configuration. The PIX Firewall overwrites the packet with the translated IP addresses.

To enable this feature, use the following command:

[no] **fixup protocol icmp error**

When disabled (as is the case with any version before 6.3), the PIX Firewall does not create xlates for intermediate nodes that generate ICMP error messages. ICMP error messages generated by the intermediate nodes between the inside host and the PIX Firewall reach the outside host without consuming any additional NAT resource. This is undesirable when an outside host uses the **traceroute** command to trace the hops to the destination on the inside of the PIX Firewall. When the PIX Firewall does not NAT the intermediate hops, all the intermediate hops appear with the translated destination IP address.

IPSec

PIX Firewall Version 6.3 provides improved support for application inspection of Encapsulating Security Payload (ESP) and for using IPSec with NAT.

ESP is an IPSec protocol that provides data confidentiality, data integrity, and protection services, optional data origin authentication, and anti-replay services. ESP encapsulates the data to be protected.

However, because ESP packets do not identify the ports that are involved, PAT is performed by assigning port 0 (zero). Only one ESP tunnel is supported at a time. Also, when the PIX Firewall has this feature enabled, it cannot terminate VPN tunnels in relation to other IPSec peers.

Application inspection of ESP traffic is disabled by default. To enable this feature, enter the following command:

```
fixup protocol esp-ike
```

When this feature is enabled, PIX Firewall preserves the IKE source port. Support is not provided for the following:

- ESP tunnel serialization
- SPI matching
- Recording of SPIs for each ESP connection

PPTP

The Point-to-Point Tunneling Protocol (PPTP) is a protocol for tunneling PPP traffic. A PPTP session is composed of one TCP channel and usually two PPTP GRE tunnels. The TCP channel is the control channel used for negotiating and managing the PPTP GRE tunnels. The GRE tunnels carries PPP sessions between the two hosts.

As described in RFC 2637, the PPTP protocol is mainly used for the tunneling of PPP sessions initiated from a modem bank PAC (PPTP Access Concentrator) to the headend PNS (PPTP Network Server). When used this way, the PAC is the remote client and the PNS is the server.

However, when used for VPN by Windows, the interaction is inverted. The PNS is a remote single-user PC that initiates connection to the head-end PAC to gain access to a central network.

PPTP application inspection is disabled by default. You use the **fixup** command to enable PPTP. The command syntax is as follows:

[no] fixup protocol pptp *1723*

When enabled, PPTP application inspection inspects PPTP protocol packets and dynamically creates the GRE connections and xlates necessary to permit PPTP traffic. Only Version 1, as defined in RFC 2637, is supported.

PAT is only performed for the modified version of GRE [RFC 2637] when negotiated over the PPTP TCP control channel. Port Address Translation is *not* performed for the unmodified version of GRE [RFC 1701, RFC 1702].

To view the xlates used by PPTP connections, enter the following command:

```
show xlate
```

This command includes output for GRE connection. PAT type is shown with the **detail** option. A string is shown for each GRE xlate. For example:

```
GRE PAT from inside:10.2.1.51/1723 to outside:192.150.49.100/0 flags ri
```

To view the status of GRE connections, enter one of the following commands:

```
show conn fport 1723
show conn lport 1723
```

You can use the **show local-host** command to display both GRE xlate and GRE connection status.

SMTP

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

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.

TFTP

Trivial File Transfer Protocol (TFTP), described in RFC1350, is a simple protocol to read and write files between a TFTP server and client. Previous to PIX Firewall Version 6.3(2), the protocol was handled with a built-in rule that permits all UDP connections from a TFTP server back to a client source port if there was a TFTP connection between the server and client.

The **fixup protocol tftp** command enhances the built-in offers several advantages over an implicit rule. The advantages of using TFTP application inspection over an implicit rule are:

- DoS prevention—To prevent a host from opening many invalid connections, a secondary channel is not created if there is an existing incomplete connection between the two hosts. This restriction dictates a client can spoof at most one request.
- Penetration prevention—When TFTP request a read or write request, a secondary channel must be opened, and traffic using the secondary channel must be initiated from the server. This restriction prevents the client from creating the secondary connection and then using that connection.
- Configurable—The **fixup protocol tftp** command can be disabled if needed.

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.

TFTP application inspection enforces the following characteristics on the secondary channel. 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.

Note Note: The **fixup protocol tftp command** is enabled by default.

TFTP Fixup must be enabled if static PAT is used to redirect TFTP traffic.

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.

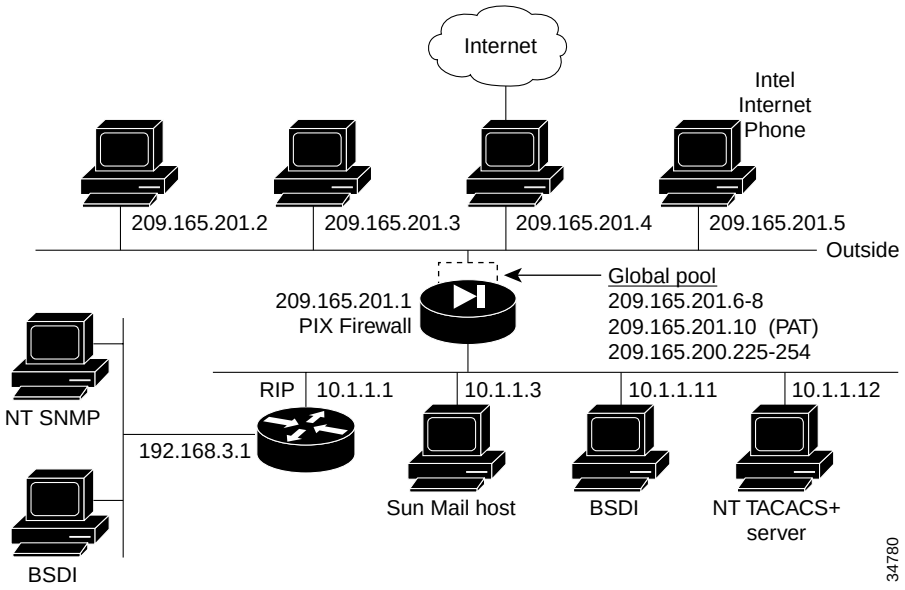
SMTP inspection monitors the command and response sequence for the following anomalous signatures:

- 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 5-3 illustrates a network scenario implementing SMTP and NFS on an internal network.

Figure 5-3 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 255.255.255.255 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 5-1 shows a command listing for configuring access to services for the network.

Example 5-1 Configuring Mail Server Access

```
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
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 255.255.255.255 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:

CTIQBE

The Telephony Application Programming Interface (TAPI) and Java Telephony Application Programming Interface (JTAPI) are used by many Cisco VoIP applications. PIX Firewall Version 6.3 introduces support for a specific protocol, Computer Telephony Interface Quick Buffer Encoding (CTIQBE), which is used by Cisco TAPI Service Provider (TSP) to communicate with Cisco CallManager.

Support for this protocol is disabled by default. To enable support for this protocol, enter the following command:

```
fixup protocol ctiqbe 2748
```

To view the status of CTIQBE connections, enter the following command:

```
show conn state ctiqbe
```

This command displays info about the media connections allocated by CTIQBE Fixup module.

In the output, the media connections allocated by CTIQBE Fixup module are denoted by a ‘C’ flag.

The following summarizes limitations that apply when using CTIQBE application inspection:

- CTIQBE application inspection does not support configurations using the **alias** command, which is deprecated after the introduction of outside NAT with PIX Firewall Version 6.2.
- Stateful Failover of CTIQBE calls is *not* supported.
- Using the **debug ctiqbe** command may delay message transmission, which may have a performance impact in a real-time environment. When you enable this debugging or logging and Cisco IP SoftPhone seems unable to complete call setup through the PIX Firewall, increase the timeout values in the Cisco TSP settings on the system running Cisco IP SoftPhone.
- CTIQBE application inspection does *not* support CTIQBE message fragmented in multiple TCP packets.

The following summarizes special considerations when using CTIQBE application inspection in specific scenarios:

- If two Cisco IP SoftPhones are registered with different Cisco CallManagers, which are connected to different interfaces of a PIX Firewall, calls between these two phones will fail.
- When Cisco CallManager is located on the higher security interface compared to Cisco IP SoftPhones, if NAT or outside NAT is required for the Cisco CallManager IP address, the mapping must be static as Cisco IP SoftPhone requires the Cisco CallManager IP address to be specified explicitly in its Cisco TSP configuration on the PC.
- When using PAT or Outside PAT, if the Cisco CallManager IP address is to be translated, its TCP port 2748 must be statically mapped to the **same port** of the PAT (interface) address in order for Cisco IP SoftPhone registrations to succeed. The CTIQBE listening port (TCP 2748) is fixed and is not user-configurable on Cisco CallManager, Cisco IP SoftPhone, or Cisco TSP.

To display information regarding the CTIQBE sessions established across the PIX Firewall, enter the following command:

```
show ctiqbe
```

For further information about using this command to troubleshoot CTIQBE application inspection issues, refer to the **show ctiqbe** command in the *Cisco PIX Firewall Command Reference*.

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.

H.323

This section describes how to manage application inspection for the H.323 suite of protocols. It includes the following topics:

Overview

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 H.323-compliant endpoints. PIX Firewall Version 5.3 through Version 6.2 supports H.323 Version 2. PIX Firewall Version 6.3 supports H.323 Version 3 and Version 4.

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

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. It has been observed that 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.

Multiple Calls on One Call Signalling Connection

PIX Firewall Version 6.3 supports multiple calls on the same call signaling channel, a feature introduced with H.323 Version 3. This feature reduces call setup time and reduces the use of ports on the PIX Firewall. A new **timeout** command is introduced to control how long the H.225 call signaling channel stays open when using this feature. The syntax for this command is as follows:

timeout h225 *hh*[*mm*[*ss*]]

Replace *hh* with the number of hours, *mm* with the minutes and *ss* with the seconds. The default is 1 hour. To keep the channel open without any timeout, set the timer to 0 by entering the following command:

```
timeout h225 00:00:00
```

To disable the timer and close the TCP connection immediately after all calls are cleared, set the timeout value to 1 second, as follows:

```
timeout h225 00:00:01
```

Viewing Connection Status

To display the status of H.225 connections, enter the following command:

```
show conn state h225
```

Technical Background

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 higher. 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 H.323 gatekeeper is in use, the initial packet is transmitted using UDP.

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 endpoints 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 an access list 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.

MGCP

Cisco PIX Firewall Version 6.3 introduces support for application inspection of the Media Gateway Control Protocol (MGCP). This section describes how to enable application inspection and view application inspection information. It includes the following topics:

Overview

MGCP is used for controlling media gateways from external call control elements called media gateway controllers or call agents. A media gateway is typically a network element that provides conversion between the audio signals carried on telephone circuits and data packets carried over the Internet or over other packet networks. Examples of media gateways are:

- Trunking gateways, that interface between the telephone network and a Voice over IP network. Such gateways typically manage a large number of digital circuits.
- Residential gateways, that provide a traditional analog (RJ11) interface to a Voice over IP network. Examples of residential gateways include cable modem/cable set-top boxes, xDSL devices, broad-band wireless devices.
- Business gateways, that provide a traditional digital PBX interface or an integrated soft PBX interface to a Voice over IP network.

MGCP messages are transmitted over UDP. A response is sent back to the source address (IP address and UDP port number) of the command, but the response may not arrive from the same address as the command was sent to. This can happen when multiple call agents are being used in a failover configuration and the call agent that received the command has passed control to a backup call agent, which then sends the response.

Enabling MGCP Application Inspection

Enter the following command to enable application inspection for MGCP:

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

Application inspection for MGCP is disabled by default. To use MGCP, you typically need to configure at least two ports. One on which the gateway receives commands and one for the port on which the call agent receives commands. Normally, a call agent will send commands to port 2427, while a gateway will send commands to port 2727.

Neither NAT or PAT are supported by PIX Firewall Version 6.3 and lower.

To enable MGCP application inspection for call agents and gateways using the default ports, enter the following commands:

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

Enter the following command to set the duration for the MGCP inactivity timer:

```
timeout mgcp hh[mm[ss]]
```

When the specified time elapses, the MGCP media ports are closed. The default is 5 minutes.

Note Enabling or changing the MGCP application inspection will have no effect until you reload the PIX Firewall configuration.

Configuration for Multiple Call Agents and Gateways

Use the following commands to configure the PIX Firewall to support the use of multiple MGCP call agents and gateways:

```
[no] mgcp call-agent ip_address group_id
[no] mgcp command-queue limit
[no] mgcp gateway ip_address group_id
```

Use the **mgcp call-agent** command to specify a group of call agents which can manage one or more gateways. This information will be used to open connections for the call agents other than the one a gateway sends a command to so that any of the call agents can send the response. The *ip_address* option specifies the IP address of the call agent. The *group_id* option is a number from 0 to 4294967295. Call agents with the same *group_id* belong to the same group.

Use the **mgcp command-queue** command to specify the maximum number of MGCP commands that will be queued waiting for a response The range of allowed values for the *limit* option is 1 to 4294967295. The default is 200. When the limit has been reached and a new command arrives, the command that has been in the queue for the longest time will be removed.

Use the **mgcp gateway** command to specify which group of call agents are managing a particular gateway. The IP address of the gateway is specified with the *ip_address* option. The *group_id* option is a number from 0 to 4294967295. It must correspond with the *group_id* of the call agents that are managing the gateway.

Use the **clear mgcp** command to remove all of the MGCP configuration and set the command queue limit to the default of 200.

The following example limits the MGCP command queue to 150 commands, allows call agents 10.10.11.5 and 10.10.11.6 to control gateway 10.10.10.115 and allows call agents 10.10.11.7 and 10.10.11.8 to control gateway 10.10.10.116:

```
mgcp call-agent 10.10.11.5 101
mgcp call-agent 10.10.11.6 101
mgcp call-agent 10.10.11.7 102
mgcp call-agent 10.10.11.8 102
mgcp command-queue 150
mgcp gateway 10.10.10.115 101
mgcp gateway 10.10.10.116 102
```

Viewing MGCP Information

To view information about MGCP, enter the following command:

```
show mgcp commands | sessions [detail]
```

Use the **commands** option to list the commands in the command queue. Use the **sessions** option to list the existing MGCP sessions. Use the **detail** option to list detailed information about each command or session.

To show information about the MGCP connections, enter the following command:

```
show conn detail |state mgcp
```

Use the **detail** option to display detailed information about the MGCP connections. Use the **state** option to display the media connections created for MGCP sessions.

SCCP

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

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 3.3. 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*]]

There are 5 versions of the SCCP protocol: 2.4, 3.0.4, 3.1.1, 3.2, and 3.3.2. PIX Firewall Version 6.3 supports up to Version 3.3.2.

Application inspection for SCCP is enabled by default. 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.

If the address of a Cisco CallManager server is configured for NAT or PAT to a different address or port and outside phones register to it using TFTP, the connection will fail because PIX Firewall does not support NAT or PAT of the file content transferred using TFTP. Although PIX Firewall does support NAT of TFTP messages and opens a pinhole for the TFTP file to traverse the firewall, PIX Firewall *cannot* translate the Cisco CallManager IP address and port embedded in the Cisco IP Phone's configuration files that are transferred using TFTP during phone registration. For a workaround to this problem, refer to the "Using SCCP with Cisco CallManager on a Higher Security Interface" section on page 5-23.

PIX Firewall Version 6.2 introduces support of DHCP options 150 and 66, 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 the PIX Firewall DCHP Server" in Chapter 4, "Using PIX Firewall in SOHO Networks."

Using PAT with SCCP

PIX Firewall Version 6.3 introduces PAT and NAT support for SCCP. PAT is necessary if you have limited numbers of global IP addresses for use by IP phones. The following are the limitations that apply to the current version of PAT and NAT support for SCCP:

- PAT will not work with configurations using the **alias** command.
- Stateful failover of SCCP calls is **not** supported.
- Use of *debug skinny* command may result in a delay of the sending of the messages which may have a performance impact in a real-time environment.
- No support for fragmented SCCP messages
- Outside NAT or PAT is **not** supported

If the **clear xlate** command is entered after PAT xlates are created for Cisco CallManager, SCCP calls cannot be established because the xlates for the Cisco CallManager are permanently deleted. Under these circumstances, Cisco IP Phones need to reregister with the Cisco CallManager to establish calls through the PIX Firewall.

In topologies where Cisco CallManager is located on the higher security interface with respect to the Cisco IP Phones, if NAT is required for the Cisco CallManager IP address, the mapping must be **static** as a Cisco IP Phone requires the Cisco CallManager IP address to be specified explicitly in its configuration.

Note If the Cisco CallManager IP address and the SCCP port must both be translated, the SCCP port must be statically mapped to the same port of the actual address for Cisco IP Phone registrations to succeed.

Using SCCP with Cisco CallManager on a 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.

When the Cisco IP Phones are on a lower security interface compared to the TFTP server, you must use an access list to connect to the protected TFTP server on UDP port 69. While you do need a static entry for the TFTP server, this does not have to be an "identity" static entry. When using NAT, an identity static entry maps to the same IP address. When using PAT, it maps to the same IP address and port.

However, if the Cisco IP Phones are on a lower security interface compared to the Cisco CallManager, we recommend that you do create an identity static entry to allow the Cisco CallManager on the higher security interface to accept registrations from the Cisco IP Phones.

When the Cisco IP Phones are on a *higher* security interface compared to the TFTP server and Cisco CallManager, no access list or static entry is required to allow the Cisco IP Phones to initiate the connection.

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 conference 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.

Viewing SCCP Information

To view information about the SCCP sessions established across the PIX Firewall, enter the following command:

```
show skinny
```

For further information about using this command to troubleshoot SCCP application inspection issues, refer to the **show skinny** command in the *Cisco PIX Firewall Command Reference*.

SIP

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

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 <udp> [*port*[**-***port*]]

Note PAT support for SIP is provided by PIX Firewall Version 6.2 or higher. Only static NAT and dynamic NAT are supported in earlier versions.

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

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 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.

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 udp 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.

Providing IP Address Privacy

Achieving IP address privacy requires the ability to retain outside IP addresses embedded in inbound SIP packets for all transactions. With the exception of REGISTER, you can hide phone IP addresses from one another by invoking ip-address privacy.

The REGISTER message and the response to REGISTER message will be exempt from this operation since this message is exchanged between the phone and the proxy.

Your can turn on this feature by using the **[no] sip ip-address-privacy** command.

Note By default this command is turned off.

When the above command is turned on, SIP fixup will retain outside IP addresses in the SIP header and SDP data of inbound SIP packets.

Here is an example of enabled IP address privacy:

```
INVITE sip:bob@Proxy SIP/2.0
Via: SIP/2.0/UDP A:5060 ================> A':patport#
From: terry@A ==========================> terry@A'
To: robin@Proxy
Call-ID:
Contact:terry@A ========================> terry@A'
SDP
o=A  ===================================> A'
c=IN IP4 A  ============================> A'
m=port#  ===============================> patport# (if applicable)

When the Proxy sends the INVITE to B:

INVITE sip:bob@Proxy SIP/2.0
Via: SIP/2.0/UDP A':5060 ============>Has to remain as A':patport#
From: terry@A' ======================>Has to remain as A'
To: robin@Proxy
Call-ID:
Contact:terry@A' ====================>Has to remain as A'
SDP
o=A'   ==============================>Has to remain as A'
c=IN IP4 A'  ========================>Has to remain as A'
m=patport#
```

If there is a requirement to hide phone ip addresses connected on the same PIX interface from each other and eliminate the direct P2P communication between the phones, this feature should be enabled. SIP ip-address-privacy managed with **fixup sip**, controls traffic (SIP) and voice (RTP/RTCP) traffic flow by creating pin holes for voice traffic. Using this feature eliminates direct point-to-point communication between phones.

Note When this feature is turned on, outside NAT/alias/bi-directional NAT and Policy NAT will not work. When a packet from the lower security level (e.g., outside) comes to the higher security level (e.g., inside), since we retain the NATted IP addresses in it, and don't send the packet through the NAT engine, outside NAT will not be performed for the inbound SIP packets.

Instant Messaging (IM)

Instant Messaging refers to the transfer of messages between users in near real-time. SIP supports the Chat feature on Windows XP using Windows Messenger RTC Client version 4.7.0105 only. The MESSAGE/INFO methods and 202 Accept response are used to support IM as defined in the following RFCs:

- Session Initiation Protocol (SIP)-Specific Event Notification, RFC 3265
- Session Initiation Protocol (SIP) Extension for Instant Messaging, RFC 3428

MESSAGE/INFO requests can come in at any time after registration/subscription. For example, two users can be online at any time, but not chat for hours. Therefore, the SIP fixup opens U_sip pinholes which will time out according to the configured SIP timeout value. This value must be configured at least five minutes longer than the subscription duration. The subscription duration is defined in the Contact Expires value and is typically 30 minutes.

Because MESSAGE/INFO requests are typically sent using a dynamically allocated port other than port 5060, they are required to go through the SIP fixup.

Note Only the Chat feature is currently supported. Whiteboard, File Transfer, and Application Sharing are not supported. RTC Client 5.0 is not supported.

Viewing SIP Information

To view information about the SIP sessions established across the PIX Firewall, enter the following command:

```
show sip
```

For further information about using this command to troubleshoot CTIQBE application inspection issues, refer to the **show sip** command in the *Cisco PIX Firewall Command Reference*.

Technical Background

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.

Note When using PAT, if a SIP device transmits a packet in which the SDP portion has an IP address in the owner/creator (o=) field that is different than the IP address in the connection field (c=), the IP address in the o= field may not be properly translated. This is due to a limitation in the SIP protocol, which does not provide a port value in the o= field.

SIP inspection has a database with indices CALL_ID/FROM/TO from the SIP payload that identifies the call, as well as the source and destination. 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.

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:

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. This section describes the different streams and includes the following topics.

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.

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, and 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.

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:

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.

By default, **fixup protocol ils** is disabled. You can use the **fixup** command to enable the ILS fixup and, optionally, change the default port assignment. 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

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 an 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 5-3. 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 5-2 shows the command listing for configuring access to services for the network illustrated in Figure 5-3.

Example 5-2 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)

The SQL*Net protocol consists of different packet types that PIX Firewall handles to make the data stream appear consistent to the Oracle applications on either side of the firewall. 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.

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

The ICMP payload is scanned to retrieve the five-tuple from the original packet. ICMP inspection supports both one-to-one 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 is updated

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. XDCMP inspection does not support PAT.

Simple Network Management Protocol Fixup

SNMP fixup enables packet traffic monitoring between network devices. Using the **fixup protocol snmp command**, the PIX Firewall can be configured to deny traffic based on packet version.

The fixup can be enabled or disabled via the fixup command **[no] fixup protocol snmp 161-162.** However, existing connections will retain the fixup configuration present when the connection was created. Use **clear xlate** or **clear local** to clear connections and allow any new fixup configuration to take effect.