Configuring Application Protocol Inspection

This chapter describes how to use and configure application protocol inspection, which is often called a "fixup." Inspection engines are required for services that embed IP addressing information in the user data packet or that open secondary channels on dynamically assigned ports. These protocols require the Firewall Services Module (FWSM) to do a deep packet inspection instead of passing the packet through the fast path (see the "Stateful Inspection Feature" section on page 1-5 for more information about the fast path). As a result, inspection engines can affect overall throughput.

Several common inspection engines are enabled on the FWSM by default, but you might need to enable others depending on your network. This chapter includes the following sections:

- Inspection Engine Overview, page 13-1
- Configuring an Inspection Engine, page 13-4
- Detailed Information About Inspection Engines, page 13-5

Inspection Engine Overview

This section includes the following topics:

- When to Use Application Protocol Inspection, page 13-1
- Inspection Limitations, page 13-2
- Inspection Support, page 13-2

When to Use Application Protocol Inspection

When a user establishes a connection, the FWSM checks the packet against access control lists (ACLs), creates an address translation, and creates an entry for the session in the fast path, so that further packets can bypass time-consuming checks. However, the fast path relies on predictable port numbers and does not perform address translations inside a packet.

Many protocols open secondary TCP or UDP ports. The initial session on a well-known port is used to negotiate dynamically assigned port numbers.

Other applications embed an IP address in the packet that needs to match the source address that is normally translated when it goes through the FWSM.

If you use applications like these, then you need to enable application inspection.

When you enable application inspection for a service that embeds IP addresses, the FWSM translates embedded addresses and updates any checksum or other fields that are affected by the translation.

When you enable application inspection for a service that uses dynamically assigned ports, the FWSM monitors sessions to identify the dynamic port assignments, and permits data exchange on these ports for the duration of the specific session.

Inspection Limitations

See the following limitations for application protocol inspection:

- You can configure up to 32 inspection engines per context. This limit includes the following inspection engines that are enabled by default, making the total number of configurable inspection engines 27: TFTP, Sun RPC over UDP, NetBIOS NameServer, XDMCP, and CUSeeMe. The OraServ and RealAudio inspection engines, which are also enabled by default, do not affect this limit.

- State information for multimedia sessions that require inspection are not passed over the state link for stateful failover.

- For fragmented IP packets, only the first fragment is inspected.

- For segmented TCP packets, if messages are divided between segments, the FWSM cannot inspect the packets.

- Some inspection engines do not support PAT, NAT, policy NAT, outside NAT, or NAT between same security interfaces. See "Inspection Support" for more information about NAT support.

Inspection Support

Table 13-1 describes the inspection engines supported by the FWSM and whether they are compatible with Network Address Translation (NAT), Port Address Translation (PAT), outside NAT, or NAT between same security interfaces. If a inspection engine does not support outside NAT, consider using the **alias** command instead of outside NAT. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information about the **alias** command.

Inspection engines that are enabled for the default port by default are in bold.

Table 13-1   Inspection Engine Support

Application[1]	Configurable	Default Port	NAT Limitations	Comments	Standards[2]
CUSeeMe	No	UDP/7648	No NAT or PAT. Use NAT identity or NAT exemption only.	—	—
DNS over UDP	Yes	UDP/53	No NAT support is available for name resolution through WINS.	No PTR records are changed.	RFC 1123
FTP	Yes	TCP/21	—	—	RFC 1123
H.323 H.225 and **RAS**	Yes	TCP/1720 UDP/1718-1719	No outside NAT. Use the **alias** command. No NAT on same security interfaces.	Does not support segmented messages.	ITU-T H.323, H.245, H225.0, Q.931, Q.932
HTTP	Yes	TCP/80	—	—	RFC 2616
ICMP	Yes	—	—	—	—
ICMP error	Yes	—	—	—	—

Table 13-1   Inspection Engine Support (continued)

Application[1]	Configurable	Default Port	NAT Limitations	Comments	Standards[2]
ILS (LDAP)	Yes	TCP/389	No outside NAT. Use the **alias** command. No PAT.	—	—
MGCP	Yes	UDP/2427, 2727	No NAT or PAT. Use NAT identity or NAT exemption only.	—	RFC2705bis-05
NetBIOS Name Server over IP	No	UDP/137-138	—	—	—
OraServ	No	UDP/1525	—	—	—
RealAudio	No	UDP/7070	—	—	—
RSH	Yes	TCP/514	No PAT.	—	Berkeley UNIX
RTSP	Yes	TCP/554	No PAT. No outside NAT. Use the **alias** command.	No handling for HTTP cloaking.	RFC 2326, RFC 2327, RFC 1889
SIP TCP	Yes	TCP/5060	No outside NAT. Use the **alias** command. No NAT on same security interfaces.	—	RFC 2543
SIP UDP	Yes	UDP/5060	No outside NAT. Use the **alias** command. No NAT on same security interfaces.	—	RFC 2543
SKINNY (SCCP)	Yes	TCP/2000	No outside NAT. Use the **alias** command. No NAT on same security interfaces.	Does not handle TFTP uploaded Cisco IP Phone configurations.	—
SMTP	Yes	TCP/25	—	—	RFC 821, 1123
SQL*Net	Yes	TCP/1521 (v1)	No policy NAT.	v1 and v2.	—
Sun RPC over UDP	No	UDP/111	No NAT or PAT. Use NAT identity or NAT exemption only.	—	—
Sun RPC over TCP	Yes	TCP/111	No NAT or PAT. Use NAT identity or NAT exemption only.	—	—
TFTP	No	UDP/69	Payload IP address not translated.	—	RFC 1350
XDMCP	No	UDP/177	No NAT or PAT. Use NAT identity or NAT exemption only.	—	—

1. Inspection engines that are enabled by default for the default port are in bold.

2. The FWSM is in compliance with these standards, but it does not enforce compliance on packets being inspected. For example, FTP commands are supposed to be in a particular order, but the FWSM does not enforce the order.

Configuring an Inspection Engine

Disabling or modifying an inspection engine only affects connections that are initiated after the command is processed. Disabling an inspection engine 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 sessions.

To configure an inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol {
dns [maximum-length length]
ftp [strict] [port[-port]] |
h323 {h225 | ras} [port[-port]] |
http [port[-port]] |
icmp |
icmp error |
ils [port[-port]] |
mgcp [port[-port] |
rpc [port[-port] |
rsh [port[-port] |
rtsp [port[-port] |
sip [port[-port] |
sip udp |
skinny [port[-port] |
smtp [port[-port]] |
sqlnet [port[-port]]}
```

For most applications and protocols, you can define multiple port assignments, which is useful when multiple instances of the same service are running on different ports.

Because you can enter multiple ports (either as a range or as separate commands), if you specify a new port, that port is added to the configuration along with previously configured ports. To remove a port, enter the **no** version of the command.

See the following keywords:

- **dns maximum-length** *length*—This option sets the maximum length of a DNS reply. The default is 512 bytes. This inspection engine uses UDP port 53, and the port is not configurable.

- **ftp strict**—This 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. This limitation prevents clients from sending embedded commands in FTP requests. Each FTP command must be acknowledged before a new command is allowed.

- **h323** {**h225** | **ras**}—You can set the inspection engines for H.323 and RAS (**h225** and **ras**) separately.

See the "Detailed Information About Inspection Engines" section on page 13-5 for information about each protocol inspection engine.

By default, an inspection engine for FTP port 21 is enabled. The following example shows how to define additional ports for FTP:

```
FWSM/contexta(config)# fixup protocol ftp 2100
FWSM/contexta(config)# fixup protocol ftp 4254
FWSM/contexta(config)# fixup protocol ftp 9090
```

After entering these commands, the FWSM listens for FTP traffic on port 21, as well as 2100, 4254, and 9090.

The following command assigns the port range from 1500 to 2000 to SQL*Net:

```
FWSM/contexta(config)# fixup protocol sqlnet 1500-2000
```

Detailed Information About Inspection Engines

CUSeeMe Inspection Engine

Enabled by default for UDP port 7648

Not Configurable

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

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

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

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

DNS over UDP Inspection Engine

Enabled by default for UDP port 53

Domain Name System (DNS) requests require an inspection engine so that DNS queries are not 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. The DNS inspection engine monitors the message exchange to ensure that the ID of the DNS reply matches the ID of the DNS query. See the "DNS and NAT" section on page 9-13 for more information about how the FWSM alters the DNS payload.

This functionality is different from DNS Guard. See the "Other Protection Features" section on page 1-6 for more information about DNS Guard.

To configure the maximum length of the DNS reply, enter the following command:

```
FWSM/contexta(config)# fixup protocol dns [maximum-length length]
```

The default is 512 bytes. The port is 53 (UDP) and is not configurable.

FTP Inspection Engine

Enabled by default for TCP port 21

To configure the FTP inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol ftp [strict] port[-port]
```

The default port is 21 (TCP).

If you disable FTP inspection engines with the **no fixup protocol ftp** command, outbound users can start connections only in passive mode, and inbound users can start connections only in active mode.

The FTP inspection engine inspects the FTP sessions, and performs four tasks:

- Prepares dynamic secondary data connection—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.

- Tracks **ftp** command-response sequence—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 of 256. 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.

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

- Generates an audit trail—The FTP inspection engine generates the following system messages:
  - System message 303002 is generated for each file that is retrieved or uploaded.
  - System message 201005 is generated if the secondary dynamic channel preparation failed due to memory shortage.

- Translates embedded IP addresses—In conjunction with NAT, the FTP inspection engine translates the IP address within the application payload. This is described in detail in RFC 959.

H.323 Inspection Engine

H.323 H.225 enabled by default for TCP port 1720

H.323 RAS enabled by default for UDP ports 1718-1719

The **fixup protocol h323** command provides support for H.323-compliant endpoints. The FWSM supports H.323 Version 1, 2, 3, and 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.

This section includes the following topics:

Configuring the H.323 Inspection Engine

To configure the H.323 inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol h323 {h225 | ras} [port[-port]]
```

You can set the inspection engines for H.232 and RAS (**h225** and **ras**) separately. The default port for **h225** is 1720 (TCP), and the default ports for **ras** are 1718-1719 (UDP).

Multiple Calls on One Call Signalling Connection

Allowing multiple calls on the same call signaling channel reduces call setup time and reduces the use of ports on the FWSM.

To configure how long the H.225 call signaling channel stays open, enter the following command:

```
FWSM/contexta(config)# timeout h225 hh[:mm[:ss]]
```

The default is 1 hour.

For example, 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:

```
FWSM/contexta(config)# show conn state h225
```

Technical Background

The H.323 collection of protocols collectively can use up to two TCP connections 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 might 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.

Note In environments where an H.323 gatekeeper is in use, the initial packet is transmitted using UDP.

The H.323 inspection engine monitors the Q.931 TCP connection to determine the H.245 port number. If the H.323 terminals are not using FastConnect, the FWSM 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. The H.323 inspection engine 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. The H.323 inspection engine uses the following ports:

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

The two major functions of the H.323 inspection engine 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, FWSM uses an ASN.1 decoder to decode the H.323 messages. The H.323 inspection engine supports static NAT and dynamic NAT. It does not support NAT on same security interfaces or outside NAT.

- Dynamically allocate the negotiated H.245 and RTP/RTCP connections.

The FWSM administrator must configure an access control list (ACL) 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.

> **Note** When an H.323 gatekeeper is used, the FWSM opens an H.225 connection based on inspection of the AdmissionConfirm (ACF) message.

The FWSM dynamically allocates the H.245 channel after inspecting the H.225 messages and then "hooks up" the H.245 channel to be fixed up as well. That means whatever H.245 messages pass through the FWSM are passed through the H.245 inspection engine, 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, the FWSM must remember the TPKT length to process/decode the messages properly. FWSM keeps a data structure for each connection and that data structure contains the TPKT length for the next expected message.

If the FWSM 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 the FWSM will proxy ACK that TPKT and append a new TPKT to the H.245 message with the new length.

> **Note** The FWSM does not support TCP options in the Proxy ACK for the TPKT.

Each UDP connection with a packet going through the H.323 inspection engine 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.

HTTP Inspection Engine

The HTTP inspection engine enables the system message 304001 when an inside user issues an HTTP GET request:

%FWSM-5-304001: *user source_address* Accessed [JAVA] URL *dest_address*: *url*.

To configure the HTTP inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol http [port[-port]]
```

The default port is 80 (TCP).

ICMP Inspection Engine

The ICMP inspection engine allows ICMP traffic to have a "session" so it can be inspected like TCP and UDP traffic. Without the ICMP inspection engine, we recommend that you do not allow ICMP through the FWSM in an ACL. Without stateful inspection, ICMP can be used to attack your network. The ICMP inspection engine ensures that there is only one response for each request, and that the sequence number is correct.

To configure the ICMP inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol icmp
```

The ICMP payload is scanned to retrieve the five-tuple from the original packet. The ICMP inspection engine 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. The ICMP inspection engine makes the following changes to the ICMP packet:

- In the IP Header, the NAT IP is changed to the Client IP (Destination Address) and the IP checksum is modified.
- In the 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 the Client IP
    - Original packet NAT port is changed to the Client Port
    - Original packet IP checksum is recalculated

ICMP Error Inspection Engine

The FWSM supports NAT of ICMP error messages. When this feature is enabled, the FWSM creates translation sessions for intermediate hops that send ICMP error messages, based on the NAT configuration. The FWSM overwrites the packet with the translated IP addresses.

To configure the ICMP error inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol icmp error
```

When disabled, the FWSM does not create translation sessions for intermediate nodes that generate ICMP error messages. ICMP error messages generated by the intermediate nodes between the inside host and the FWSM 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 FWSM. When the FWSM does not NAT the intermediate hops, all the intermediate hops appear with the translated destination IP address.

ILS Inspection Engine

Enabled by default for TCP port 389

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.

To configure the ILS inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol ils [port[-port]]
```

The default port is 389 (TCP).

The FWSM supports NAT for ILS, which is used to register and locate endpoints in the ILS or SiteServer Directory. PAT is not supported because only IP addresses, and not ports, 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, translation sessions are searched first, and then NAT entries to obtain the correct address. If both of these searches fail, then the address is not changed.

Note For sites using NAT exemption or identity NAT, we recommend that you disable this inspection engine for better performance.

Additional configuration might be necessary when the ILS server is located inside the FWSM border. This requires an ACL 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 might be created.

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

The ILS inspection engine performs the following operations:

- Decodes the LDAP REQUEST/RESPONSE PDUs using the bit error rate (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

The ILS inspection engine has the following limitations:

- Referral requests and responses are not supported
- Users in multiple directories are not unified
- Single hosts that register to multiple directories using different name are not supported by the ILS inspection engine. You must use the same for all directories.

MGCP Inspection Engine

The Media Gateway Control Protocol (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.

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 sends commands to port 2427, while a gateway sends commands to port 2727.

To configure the MGCP inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol mgcp port[-port]
```

The default ports are 2427 and 2727

Neither NAT or PAT are supported by the FWSM with MGCP.

This section includes the following topics:

MGCP Overview

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.

Configuration for Multiple Call Agents and Gateways

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

- To specify a group of call agents that can manage one or more gateways, enter the following command:

```
FWSM/contexta(config)# mgcp call-agent ip_address group_id
```

This information is used to open connections for the other call agents than the one a gateway sends a command to, so that any of the call agents can send the response. The *ip_address* specifies the IP address of the call agent. The *group_id* is a number from 0 to 4294967295. Call agents with the same *group_id* belong to the same group.

- To specify the maximum number of MGCP commands that can be queued waiting for a response, enter the following command:

```
FWSM/contexta(config)# mgcp command-queue limit
```

The range of allowed values for *limit* is 1 to 4294967295. The default is 200. When the limit is reached and a new command arrives, the command that was in the queue for the longest time is removed.

- To specify which group of call agents are managing a particular gateway, enter the following command:

```
FWSM/contexta(config)# mgcp gateway ip_address group_id
```

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.

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:

```
FWSM/contexta(config)# mgcp call-agent 10.10.11.5 101
FWSM/contexta(config)# mgcp call-agent 10.10.11.6 101
FWSM/contexta(config)# mgcp call-agent 10.10.11.7 102
FWSM/contexta(config)# mgcp call-agent 10.10.11.8 102
```

```
FWSM/contexta(config)# mgcp command-queue 150
FWSM/contexta(config)# mgcp gateway 10.10.10.115 101
FWSM/contexta(config)# mgcp gateway 10.10.10.116 102
```

Viewing MGCP Information

- To view information about MGCP, enter the following command:

  FWSM/contexta(config)# **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:

  FWSM/contexta(config)# **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.

NetBios Name Service Inspection Engine

Enabled by default for UDP ports 137 and 138

Not Configurable

The NetBios inspection engine translates IP addresses in the NetBios name service (NBNS) packets according to the FWSM NAT configuration.

OraServ Inspection Engine

Enabled by default for UDP port 1525

Not Configurable

The OraServ inspection engine allows the data channel to go through the FWSM.

RealAudio Inspection Engine

Enabled by default for UDP port 7070

Not Configurable

The RealAudio inspection engine allows the data channel to go through the FWSM when the data channel source port is between UDP ports 6790 and 7170.

RSH Inspection Engine

Enabled by default for TCP port 514

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

To configure the RSH inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol rsh [port[-port]]
```

The default port for the initial RSH connection is 514 (TCP).

RTSP Inspection Engine

Real Time Streaming Protocol (RTSP) is used by RealAudio, RealNetworks, Apple QuickTime 4, RealPlayer, and Cisco IP/TV connections. FWSM does not support multicast RTSP.

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

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

The supported Real Data Transports (RDTs) are: rtp/avp, rtp/avp/udp, x-real-rdt, x-real-rdt/udp, and x-pn-tng/udp.

The FWSM parses Setup response messages with a status code of 200. If the response message is travelling inbound, the server is outside relative to the FWSM and dynamic channels need to be opened for connections coming inbound from the server. If the response message is outbound, then the FWSM 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 FWSM 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.

To configure the RTSP inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol rtsp [port[-port]]
```

The default port is 554 (TCP).

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

```
FWSM/contexta(config)# fixup protocol rtsp 554
FWSM/contexta(config)# fixup protocol rtsp 8554
```

The following restrictions apply to the RTSP inspection engine:

- The FWSM does not inspect RTSP messages passing through UDP ports.
- The FWSM does not inspect inbound RTSP connections.

- The FWSM does not support RealNetworks multicast mode (x-real-rdt/mcast).

- The FWSM does not support PAT and outside NAT for RTSP.

- The FWSM does not have the ability to recognize HTTP cloaking where RTSP messages are hidden in HTTP messages.

- The FWSM cannot perform NAT on RTSP messages because the embedded IP addresses are contained in the Session Description Protocol (SDP) files as part of HTTP or RTSP messages. Packets could be fragmented, and the FWSM cannot perform NAT on fragmented packets.

- With Cisco IP/TV, the number of translations the FWSM 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 FWSM, 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 you use 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 FWSM, there is no need to configure the inspection engine.

  If you use 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 FWSM, configure the RTSP inspection engine.

SIP Inspection Engine

Enabled by default for TCP and UDP port 5060

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 includes the following topics:

Configuring the SIP Inspection Engine

To configure the SIP inspection engine, enter the following commands:

- To configure the SIP TCP inspection engine, enter the following command:

  ```
  FWSM/contexta(config)# fixup protocol sip [port[-port]]
  ```

  The default port is 5060 (TCP).

- To configure the SIP UDP inspection engine, enter the following command:

  ```
  FWSM/contexta(config)# fixup protocol sip udp
  ```

  The default port is 5060 (UDP), which is the only port allowed.

SIP Overview

SIP works with Session Description Protocol (SDP) for call signalling. SDP specifies the ports for the media stream. The inspection engine supports the following SIP message types. Other message types are allowed through the FWSM, but they are not inspected.

- Messages in RFC 2543 (redefined in RFC 3261):
  - INVITE
  - ACK
  - BYE
  - CANCEL
  - REGISTER
  - Responses 1xx, 2xx, 3xx, 4xx, 5xx, 6xx
- Message in RFC 2976:
  - INFO
- Messages in RFC 3265:
  - SUBSRIBE
  - NOTIFY
- Message in RFC 3428:
  - MESSAGE

To support SIP calls through the FWSM, the FWSM inspects signaling messages for the media connection addresses, media ports, and embryonic connections for the media, 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. The SIP inspection engine applies NAT for these embedded IP addresses. It does not support NAT between same security interfaces or outside NAT.

Technical Background

The SIP inspection engine 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.

The SIP inspection engine has a database that keeps track of information 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 inspection engine 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 FWSM, unless the FWSM 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. See the **timeout** command in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

Skinny Inspection Engine

Enabled by default for TCP port 2000

Skinny (or Simple) Client Control Protocol (SCCP) is a protocol used in VoIP networks.

To configure the Skinny inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol skinny [port[-port]]
```

The default port is 2000 (TCP).

This section includes the following topics:

- Skinny Overview, page 13-18
- Problems with Fragmented Skinny Packets, page 13-19

Skinny Overview

Cisco IP Phones using Skinny can coexist with an H.323 environment. When used with Cisco CallManager, the Skinny client can interoperate with H.323-compliant terminals. The FWSM ensures that all SCCP signalling and media packets can traverse the FWSM by providing NAT of the SCCP Signaling packets. This inspection engine does not support NAT between same security interfaces.

There are 5 versions of the SCCP protocol supported: 2.4, 3.0.4, 3.1.1, 3.2, and 3.3.2.

The FWSM supports DHCP options 150 and 66, which allow the FWSM to send the location of a TFTP server to Cisco IP Phones and other DHCP clients. The TFTP server provides the address of the Cisco CallManager for the Cisco IP Phones. For further information about this feature, see the "Configuring the DHCP Server" section on page 8-19. If the Cisco CallManager is on a higher security interface, which requires NAT for the Cisco CallManager IP address, and you configure the TFTP server to serve a file with the local untranslated address of the Cisco CallManager, then the Cisco IP Phones cannot contact the Cisco CallManager. We recommend that you use the Cisco CallManager name instead of the IP address, and rely on the DNS server to provide the correct address. If the DNS server is also on the higher security interface, the FWSM can use the DNS inspection engine to translate the address inside the DNS response.

If you enter the **clear xlate** command after PAT translations are created for Cisco CallManager, Skinny calls cannot be established because the translations 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 FWSM.

Problems with Fragmented Skinny Packets

The FWSM does not correctly handle fragmented Skinny packets. For instance, when using a voice conferencing bridge, Skinny packets might become fragmented and are then dropped by the FWSM. This happens because the Skinny inspection engine checks each packet and drops what appear to be bad packets. When a single Skinny packet is fragmented into multiple TCP packets, the Skinny inspection engine finds that the internal checksums within the Skinny packet fragments are not correct and so it drops the packet.

SMTP Inspection Engine

Enabled by default for TCP port 25

The SMTP inspection engine 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. The FWSM converts 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 might cause Microsoft Outlook clients and Exchange servers to function unpredictably when their connection passes through FWSM. In this case, you might want to disable the SMTP inspection engine, although the Mail Guard feature does provide valuable protection.

To configure the SMTP inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol smtp [port[-port]]
```

The default port is 25 (TCP).

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

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

The SMTP inspection engine 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 FWSM 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 packet, the TCP checksum has to be recalculated.

- TCP stream editing.

- Command pipelining.

SQL*Net Inspection Engine

Enabled by default for TCP port 1521

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

To configure the SQL*Net inspection engine, enter the following command:

```
FWSM/contexta(config)# fixup protocol sqlnet [port[-port]]
```

The default port is 1521 (TCP).

The FWSM 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 are fixed up.

The packets that need inspection engine 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) are not scanned for addresses to NAT, nor does the inspection engine open dynamic connections for any embedded ports in the packet.

SQL*Net Version 2 TNSFrames, Redirect, and Data packets are 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 FWSM, a flag is set in the connection data structure to expect the Data or Redirect message that follows is NATed and ports are dynamically opened. If one of the TNS frames in the preceding paragraph arrives after the Redirect message, the flag is reset.

The SQL*Net inspection engine recalculates the checksum, change IP, TCP lengths, and readjusts 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 are scanned for ports and addresses. Addresses are NATed and port connections are opened.

Sun RPC Inspection Engine

Enabled by default for UDP port 111

Sun Remote Procedure Call (RPC) is used by many services, for example, 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 sends its RPC queries to that new port.

When a server sends out a reply, the FWSM intercepts this packet and opens both embryonic TCP and UDP connections on that port for a short period of time. After the client connects to the port and makes a full connection, the embryonic connection goes away. For additional connections from the client to the port, the client must repeat the portmapper process. Alternatively, you can configure the FWSM to keep the embryonic connections open for a longer period of time so that clients can use cached port numbers and do not have to repeat the portmapper process. This method is required for Sun RPC over TCP; only the default inspection for UDP uses the above method. See the **rpc-server** command below.

NAT or PAT of RPC payload information is not supported. Use NAT exemption or identity NAT.

- To configure the Sun RPC inspection engine for TCP, enter the following command:

  FWSM/contexta(config)# **fixup protocol rpc** [*port*[*-port*]]

  The default port is 111 (TCP). You must also configure the **rpc-server** command (below). The UDP inspection engine is on by default and is not configurable.

- To allow clients to use cached port numbers for Sun RPC services (such as NFS or NIS), enter the following command:

  FWSM/contexta(config)# **rpc-server** *interface_name ip_address mask* **service** *service_type* **protocol** {**tcp** | **udp**} *port*[*-port*] **timeout** *hh:mm:ss*

  After a client initially connects to a server running a Sun RPC service, the client might cache the Sun RPC port information supplied by the portmapper process. Additional connections from the client might use these cached ports. This command allows clients to use cached port numbers for the duration of the specified **timeout** rather than have to re-request the port numbers from the portmapper process. This command is required for Sun RPC over TCP.

TFTP Inspection Engine

Enabled by default for UDP port 69

Not Configurable

The FWSM permits all UDP connections from a TFTP server back to a client source port if there is an existing TFTP connection between the server and client.

XDMCP Inspection Engine

Enabled by default for UDP port 177

Not Configurable

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 as the start of an Xwindows session, the FWSM must allow the TCP back connection. Once XDMCP negotiates the session, a single embryonic connection is created to handle the initial TCP connection, after which the established rule is consulted.

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 FWSM can NAT if needed. The XDCMP inspection engine does not support PAT.