

inspect a – inspect z

- inspect ctiqbe, on page 3
- inspect dcerpc, on page 5
- inspect diameter, on page 7
- inspect dns, on page 9
- inspect esmtp, on page 11
- inspect ftp, on page 14
- inspect gtp, on page 17
- inspect h323, on page 20
- inspect http, on page 22
- inspect icmp, on page 24
- inspect icmp error, on page 26
- inspect ils, on page 28
- inspect im, on page 31
- inspect ip-options, on page 33
- inspect ipsec-pass-thru, on page 36
- inspect ipv6, on page 38
- inspect lisp, on page 40
- inspect m3ua, on page 42
- inspect mgcp, on page 44
- inspect mmp, on page 47
- inspect netbios, on page 49
- inspect pptp, on page 51
- inspect radius-accounting, on page 53
- inspect rsh, on page 55
- inspect rtsp, on page 57
- inspect scansafe, on page 60
- inspect sctp, on page 63
- inspect sip, on page 65
- inspect skinny, on page 68
- inspect snmp, on page 71
- inspect sqlnet, on page 73
- inspect stun, on page 75
- inspect sunrpc, on page 77

- inspect tftp, on page 79
- inspect vxlan, on page 81
- inspect waas, on page 83
- inspect xdmcp, on page 84

inspect ctiqbe

To enable CTIQBE protocol inspection, use the **inspect ctiqbe** command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To disable inspection, use the **no** form of this command.

inspect ctiqbe
no inspect ctiqbe

Command Default

This command is disabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added and replaces the previously existing **fixup** command, which has been deprecated.

Usage Guidelines

The **inspect ctiqbe** command enables CTIQBE protocol inspection, which supports NAT, PAT, and bidirectional NAT. This enables Cisco IP SoftPhone and other Cisco TAPI/JTAPI applications to work successfully with Cisco CallManager for call setup across the ASA .

The Telephony Application Programming Interface (TAPI) and Java Telephony Application Programming Interface (JTAPI) are used by many Cisco VoIP applications. Computer Telephony Interface Quick Buffer Encoding (CTIQBE) is used by Cisco TAPI Service Provider (TSP) to communicate with Cisco CallManager.

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

- 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 ASA, increase the timeout values in the Cisco TSP settings on the system running Cisco IP SoftPhone.
- CTIQBE application inspection does not support CTIQBE messages 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 the ASA, 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 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.

Inspecting Signaling Messages

For inspecting signaling messages, the **inspect** ctiqbe command often needs to determine locations of the media endpoints (for example, IP phones).

This information is used to prepare access control and NAT state for media traffic to traverse the firewall transparently without manual configuration.

In determining these locations, the **inspect** ctiqbe command does not use the tunnel default gateway route. A tunnel default gateway route is a route of the form **route** *interface* **0 0** *metric* **tunneled**. This route overrides the default route for packets that egress from IPsec tunnels. Therefore, if the **inspect** ctiqbe command is desired for VPN traffic, do not configure the tunnel default gateway route. Instead, use other static routing or dynamic routing.

Examples

The following example enables the CTIQBE inspection engine, which creates a class map to match CTIQBE traffic on the default port (2748). The service policy is then applied to the outside interface.

```
ciscoasa(config)# class-map ctiqbe-port
ciscoasa(config-cmap)# match port tcp eq 2748
ciscoasa(config-cmap)# exit
ciscoasa(config)# policy-map ctiqbe_policy

ciscoasa(config-pmap)# class ctiqbe-port
ciscoasa(config-pmap-c)# inspect ctiqbe
ciscoasa(config-pmap-c)# exit
ciscoasa(config)# service-policy ctiqbe_policy interface outside
```

To enable CTIQBE inspection for all interfaces, use the **global** parameter in place of **interface outside**.

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
show conn	Displays the connection state for different connection types.
show ctiqbe	Displays information regarding the CTIQBE sessions established across the ASA and the media connections allocated by the CTIQBE inspection engine.
timeout	Sets the maximum idle time duration for different protocols and session types.

inspect dcerpc

To enable inspection of DCERPC traffic destined for the endpoint-mapper, use the inspect dcerpc command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect dcerpc [*map_name*]
no inspect dceprc [*map_name*]

Syntax Description

map_name	(Optional) The name of the DCERPC inspection map.

Command Default

This command is disabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.2(1)	This command was added.

Usage Guidelines

The **inspect dcerpc** command enables or disables application inspection for the DCERPC protocol.

Examples

The following example shows how to define a DCERPC inspection policy map with the timeout configured for DCERPC pinholes.

```
ciscoasa(config)# policy-map type inspect dcerpc dcerpc_map
ciscoasa(config-pmap)# parameters

ciscoasa(config-pmap-p)# timeout pinhole 0:10:00
ciscoasa(config)# class-map dcerpc
ciscoasa(config-cmap)# match port tcp eq 135
ciscoasa(config)# policy-map global-policy
ciscoasa(config-pmap)# class dcerpc
ciscoasa(config-pmap-c)# inspect dcerpc dcerpc_map
ciscoasa(config)# service-policy global-policy global
```

Related Commands

Commands	Description
class	Identifies a class map name in the policy map.

Commands	Description
class-map type inspect	Creates an inspection class map to match traffic specific to an application.
policy-map	Creates a Layer 3/4 policy map.
policy-map type inspect	Creates an inspection policy map.
show running-config policy-map	Display all current policy map configurations.
timeout pinhole	Configures the timeout for DCERPC pinholes and overrides the global system pinhole timeout.

inspect diameter

To enable Diameter application inspection, use the inspect **diameter** command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect diameter [*diameter_map*] [**tls-proxy** *proxy_name*]
no inspect diameter [*diameter_map*] [**tls-proxy** *proxy_name*]

Note Diameter inspection requires the Carrier license.

Syntax Description

diameter_map	Specifies a Diameter policy map name.
tls-proxy *proxy_name*	Uses the specified TLS proxy so that encrypted connections can be inspected.

Command Default

This command is disabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
9.5(2)	This command was added.
9.6(1)	The **tls-proxy** keyword was added.

Usage Guidelines

Diameter is an Authentication, Authorization, and Accounting (AAA) protocol used in next-generation mobile and fixed telecom networks such as EPS (Evolved Packet System) for LTE (Long Term Evolution) and IMS (IP Multimedia Subsystem). It replaces RADIUS and TACACS in these networks.

Diameter uses TCP and SCTP as the transport layer, and secures communications using TCP/TLS and SCTP/DTLS. It can optionally provide data object encryption as well. For detailed information on Diameter, see RFC 6733.

Diameter applications perform service management tasks such as deciding user access, service authorization, quality of service, and rate of charging. Although Diameter applications can appear on many different control-plane interfaces in the LTE architecture, the ASA inspects Diameter command codes and attribute-value pairs (AVP) for the following interfaces only:

- S6a: Mobility Management Entity (MME) - Home Subscription Service (HSS).
- S9: PDN Gateway (PDG) - 3GPP AAA Proxy/Server.
- Rx: Policy Charging Rules Function (PCRF) - Call Session Control Function (CSCF).

Diameter inspection opens pinholes for Diameter endpoints to allow communication. The inspection supports 3GPP version 12 and is RFC 6733 compliant. You can use it for TCP/TLS (by specifying a TLS proxy when you enable inspection) and SCTP, but not SCTP/DTLS. Use IPsec to provide security to SCTP Diameter sessions.

You can optionally use a Diameter inspection policy map to filter traffic based on application ID, command codes, and AVP, to apply special actions such as dropping packets or connections, or logging them. You can create custom AVP for newly-registered Diameter applications. Filtering let's you fine-tune the traffic you allow on your network.



Note Diameter messages for applications that run on other interfaces will be allowed and passed through by default. However, you can configure a Diameter inspection policy map to drop these applications by application ID, although you cannot specify actions based on the command codes or AVP for these unsupported applications.

Examples

The following example applies Diameter inspection globally on the default ports, which are TCP/3868, TCP/5868, and SCTP/3868.

```
ciscoasa(config)# policy-map global_policy

ciscoasa(config-pmap)# class inspection_default
ciscoasa(config-pmap-c)# inspect diameter
ciscoasa(config-pmap-c)# exit
ciscoasa(config)# service-policy global_policy global
```

Related Commands

Commands	Description
class	Defines the traffic class to which to apply security actions.
inspect sctp	Enables SCTP inspection.
policy-map type inspect	Creates an inspection policy map.
service-policy	Applies a policy map to one or more interfaces.
show service-policy inspect diameter	Shows the status and statistics of the inspect diameter policy.
tls-proxy	Defines a TLS proxy.

inspect dns

To enable DNS inspection (if it has been previously disabled) or to configure DNS inspection parameters, use the **inspect dns** command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To disable DNS inspection, use the **no** form of this command.

inspect dns [*map_name*] [**dynamic-filter-snoop**]
no inspect dns [*map_name*] [**dynamic-filter-snoop**]

Syntax Description

dynamic-filter-snoop	(Optional) Enables dynamic filter snooping, which is used exclusively by the Botnet Traffic Filter. Include this keyword only if you use Botnet Traffic Filtering. We suggest that you enable DNS snooping only on interfaces where external DNS requests are going. Enabling DNS snooping on all UDP DNS traffic, including that going to an internal DNS server, creates unnecessary load on the ASA.
map_name	(Optional) Specifies the name of the DNS map.

Command Default

This command is enabled by default. Botnet Traffic Filter snooping is disabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added. It replaced the **fixup** command, which has been deprecated.
7.2(1)	This command was modified to allow configuration of additional DNS inspection parameters.
8.2(1)	The **dynamic-filter-snoop** keyword was added.

Usage Guidelines

DNS inspection is enabled by default, using the preset_dns_map inspection class map:

- The maximum DNS message length is 512 bytes.
- The maximum client DNS message length is automatically set to match the Resource Record.
- DNS Guard is enabled, so the ASA tears down the DNS session associated with a DNS query as soon as the DNS reply is forwarded by the ASA. The ASA also monitors the message exchange to ensure that the ID of the DNS reply matches the ID of the DNS query.
- Translation of the DNS record based on the NAT configuration is enabled.

- Protocol enforcement is enabled, which enables DNS message format check, including domain name length of no more than 255 characters, label length of 63 characters, compression, and looped pointer check.

DNS Inspection Required for DNS Rewrite

When DNS inspection is enabled, DNS rewrite provides full support for NAT of DNS messages originating from any interface.

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. If the DNS inspection engine is disabled, the A-record is not translated.

DNS rewrite performs two functions:

- Translating a public address (the routable or "mapped" address) in a *>DNS reply* to a private address (the "real" address) when the DNS client is on a private interface.
- Translating a private address to a public address when the DNS client is on the public interface.

As long as DNS inspection remains enabled, you can configure DNS rewrite for NAT.

Examples

The following example shows how to set the maximum DNS message length:

```
ciscoasa(config)# policy-map type inspect dns dns-inspect
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)# message-length maximum 1024
```

The following example creates a class map for all UDP DNS traffic, enables DNS inspection and Botnet Traffic Filter snooping with the default DNS inspection policy map, and applies it to the outside interface:

```
ciscoasa(config)# class-map dynamic-filter_snoop_class
ciscoasa(config-cmap)# match port udp eq domain
ciscoasa(config-cmap)# policy-map dynamic-filter_snoop_policy
ciscoasa(config-pmap)# class dynamic-filter_snoop_class
ciscoasa(config-pmap-c)# inspect dns preset_dns_map dynamic-filter-snoop
ciscoasa(config-pmap-c)# service-policy dynamic-filter_snoop_policy interface outside
```

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
dynamic-filter enable	Enables the Botnet Traffic Filter for a class of traffic or for all traffic if you do not specify an access list.
policy-map	Associates a class map with specific security actions.
policy-map type inspect	Creates an inspection policy map.
service-policy	Applies a policy map to one or more interfaces.

inspect esmtp

To enable SMTP/ESMTP application inspection or to change the ports to which the ASA listens, use the **inspect esmtp** command in class configuration mode. The class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect esmtp [*map_name*]
no inspect esmtp [*map_name*]

Syntax Description

map_name	(Optional) The name of the ESMTP map.

Command Default

This command is enabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added. It replaced the **fixup** command, which has been deprecated.

Usage Guidelines

ESMTP inspection is enabled by default, using the _default_esmtp_map inspection policy map.

- The server banner is masked.
- Encrypted traffic is inspected.
- Special characters in sender and receiver address are not noticed, no action is taken.
- Connections with command line length greater than 512 are dropped and logged.
- Connections with more than 100 recipients are dropped and logged.
- Messages with body length greater than 998 bytes are logged.
- Connections with header line length greater than 998 are dropped and logged.
- Messages with MIME filenames greater than 255 characters are dropped and logged.
- EHLO reply parameters matching “others” are masked.

ESMTP application inspection provides improved protection against SMTP-based attacks by restricting the types of SMTP commands that can pass through the ASA and by adding monitoring capabilities.

ESMTP is an enhancement to the SMTP protocol and is similar is most respects to SMTP. For convenience, the term SMTP is used in this document to refer to both SMTP and ESMTP. The application inspection process for extended SMTP is similar to SMTP application inspection and includes support for SMTP sessions. Most commands used in an extended SMTP session are the same as those used in an SMTP session but an ESMTP session is considerably faster and offers more options related to reliability and security, such as delivery status notification.

Extended SMTP application inspection adds support for these extended SMTP commands, including AUTH, EHLO, ETRN, HELP, SAML, SEND, SOML, STARTTLS, and VRFY. Along with the support for seven RFC 821 commands (DATA, HELO, MAIL, NOOP, QUIT, RCPT, RSET), the ASA supports a total of fifteen SMTP commands.

Other extended SMTP commands, such as ATRN, ONEX, VERB, CHUNKING, and private extensions and are not supported. Unsupported commands are translated into Xs, which are rejected by the internal server. This results in a message such as “500 Command unknown: 'XXX'.” Incomplete commands are discarded.

The ESMTP inspection engine 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.

With SMTP inspection enabled, a Telnet session used for interactive SMTP may hang if the following rules are not observed: SMTP commands must be at least four characters in length; must be terminated with carriage return and line feed; and must wait for a response before issuing the next reply.

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 basic SMTP commands and eight extended commands.
- 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.

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 ASA changes all the characters in the packet to X. In this case, the server generates an error code to the client. Because of the change in the packet, the TCP checksum has to be recalculated or adjusted.
- TCP stream editing.
- Command pipelining.

Examples

The following example enables the SMTP inspection engine, which creates a class map to match SMTP traffic on the default port (25). The service policy is then applied to the outside interface.

```
ciscoasa(config)# class-map smtp-port
ciscoasa(config-cmap)# match port tcp eq 25
ciscoasa(config-cmap)# exit
ciscoasa(config)# policy-map smtp_policy

ciscoasa(config-pmap)# class smtp-port
ciscoasa(config-pmap-c)# inspect esmtp
ciscoasa(config-pmap-c)# exit
ciscoasa(config)# service-policy smtp_policy interface outside
```

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
policy-map	Associates a class map with specific security actions.
policy-map type inspect	Creates an inspection policy map.
service-policy	Applies a policy map to one or more interfaces.
show conn	Displays the connection state for different connection types, including SMTP.

inspect ftp

To configure the port for FTP inspection or to enable enhanced inspection, use the **inspect ftp** command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect ftp [**strict** [*map_name*]]
no inspect ftp [**strict** [*map_name*]]

Syntax Description

map_name	The name of an FTP inspection map.
strict	(Optional) Enables enhanced inspection of FTP traffic and forces compliance with RFC standards.

Command Default

FTP inspection is enabled by default, and the ASA listens to port 21 for FTP.

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

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added. It replaced the **fixup** command, which has been deprecated. The *map_name* option was added.

Usage Guidelines

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

- Prepares dynamic secondary data connection
- Tracks the FTP command-response sequence
- Generates an audit trail
- Translates the embedded IP address

FTP application inspection prepares secondary channels for FTP data transfer. Ports for these channels are negotiated through PORT or PASV commands. The channels are allocated in response to a file upload, a file download, or a directory listing event.

Note Apply inspection only to the port for the FTP control connection and not the data connection. The ASA stateful inspection engine dynamically prepares the data connection as necessary.

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

Strict FTP

Strict FTP increases the security of protected networks by preventing web browsers from sending embedded commands in FTP requests. To enable strict FTP, include the strict option with the **inspect ftp** command.

When you use strict FTP, you can optionally specify an FTP inspection policy map to specify FTP commands that are not permitted to pass through the ASA.

After you enable the **strict** option on an interface, FTP inspection enforces the following behavior:

- An FTP command must be acknowledged before the ASA allows a new command.
- The ASA drops connections that send embedded commands.
- The 227 and PORT commands are checked to ensure they do not appear in an error string.

Caution Using the **strict** option may cause the failure of FTP clients that are not strictly compliant with FTP RFCs.

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—The ASA closes the connection if it detects 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.
- The ASA replaces the FTP server response to the SYST command with a series of Xs. to prevent the server from revealing its system type to FTP clients. To override this default behavior, use the **no mask-syst-reply** command in the FTP map.

FTP Log Messages

FTP application inspection generates the following log messages:

- An Audit record 302002 is generated for each file that is retrieved or uploaded.
- Audit record 201005 is generated if the secondary dynamic channel preparation failed due to memory shortage.

Examples

Before submitting a username and password, all FTP users are presented with a greeting banner. By default, this banner includes version information useful to hackers trying to identify weaknesses in a system. The following example shows how to mask this banner:

```
ciscoasa(config)# policy-map type inspect ftp mymap
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)# mask-banner
ciscoasa(config-pmap-p)# exit
ciscoasa(config-pmap)# exit
ciscoasa(config)# class-map match-all ftp-traffic
ciscoasa(config-cmap)# match port tcp eq ftp
ciscoasa(config-cmap)# exit
ciscoasa(config)# policy-map ftp-policy
ciscoasa(config-pmap)# class ftp-traffic
ciscoasa(config-pmap-c)# inspect ftp strict mymap
ciscoasa(config-pmap-c)# exit
ciscoasa(config-pmap)# exit
ciscoasa(config)# service-policy ftp-policy interface inside
```

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
mask-syst-reply	Hides the FTP server response from clients.
policy-map	Associates a class map with specific security actions.
policy-map type inspect	Creates an inspection policy map.
request-command deny	Specifies FTP commands to disallow.
service-policy	Applies a policy map to one or more interfaces.

inspect gtp

To enable GTP inspection, use the **inspect gtp** command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. Use the **no** form of this command to disable GTP inspection.

inspect gtp [*map_name*]
no inspect gtp [*map_name*]

Note GTP inspection requires the GTP/GPRS or Carrier license.

Syntax Description

map_name (Optional) Name for the GTP inspection policy map.

Command Default

This command is disabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added.
9.5(1)	Support was added for GTPv2 and IPv6 addresses.

Usage Guidelines

GPRS Tunneling Protocol is used in GSM, UMTS and LTE networks for general packet radio service (GPRS) traffic. GTP provides a tunnel control and management protocol to provide GPRS network access for a mobile station by creating, modifying, and deleting tunnels. GTP also uses a tunneling mechanism for carrying user data packets. Service provider networks use GTP to tunnel multi-protocol packets through the GPRS backbone between endpoints.

GTP inspection is not enabled by default. However, if you enable it without specifying your own inspection map, a default map is used which provides the following processing. You need to configure a map only if you want different values.

- Errors are not permitted.
- The maximum number of requests is 200.
- The maximum number of tunnels is 500.

- The GSN/endpoint timeout is 30 minutes.
- The PDP context timeout is 30 minutes. In GTPv2, this is the bearer context timeout.
- The request timeout is 1 minute.
- The signaling timeout is 30 minutes.
- The tunneling timeout is 1 hour.
- The T3 response timeout is 20 seconds.
- Unknown message IDs are dropped and logged. This behavior is confined to messages the 3GPP defines for the S5S8 interface. Messages defined for other GPRS interfaces might be allowed with minimal inspection.

Use the **policy-map type inspect gtp** command to define the parameters for GTP. After defining the GTP map, you use the **inspect gtp** command to enable the map. Then you use the **class-map**, **policy-map**, and **service-policy** commands to define a class of traffic, to apply the inspect command to the class, and to apply the policy to one or more interfaces.

The well-known ports for GTP are UDP 3386, 2123, and 2152.

Inspecting Signaling Messages

For inspecting signaling messages, the **inspect gtp** command often needs to determine locations of the media endpoints (for example, IP phones).

This information is used to prepare access control and NAT state for media traffic to traverse the firewall transparently without manual configuration.

In determining these locations, the **inspect gtp** command does **not** use the tunnel default gateway route. A tunnel default gateway route is a route of the form **route** *interface* **0 0** *metric* **tunneled**. This route overrides the default route for packets that egress from IPsec tunnels. Therefore, if the **inspect gtp** command is desired for VPN traffic, do not configure the tunnel default gateway route. Instead, use other static routing or dynamic routing.

Examples

The following example shows how to limit the number of tunnels in the network:

```
ciscoasa(config)# policy-map type inspect gtp
 gmap

ciscoasa(config-pmap)# parameters

ciscoasa(config-pmap-p)# tunnel-limit 3000

ciscoasa(config)# policy-map global_policy
ciscoasa(config-pmap)# class inspection_default

ciscoasa(config-pmap-c)# inspect gtp gmap

ciscoasa(config)# service-policy global_policy global
```

Related Commands

Commands	Description
class	Defines the traffic class to which to apply security actions.

Commands	Description
clear service-policy inspect gtp	Clears global GTP statistics.
policy-map type inspect	Creates an inspection policy map.
service-policy	Applies a policy map to one or more interfaces.
show service-policy inspect gtp	Shows that status and statistics of the inspect gtp policy.

inspect h323

To enable H.323 application inspection or to change the ports to which the ASA listens, use the **inspect h323** command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect h323 { **h225** | **ras** } [*map_name*]
no inspect h323 { **h225** | **ras** } [*map_name*]

Syntax Description

h225	Enables H.225 signaling inspection.
map_name	(Optional) The name of the H.323 map.
ras	Enables RAS inspection.

Command Default

The default port assignments are as follows:

- h323 h225 1720
- h323 ras 1718-1719

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added. It replaced the **fixup** command, which has been deprecated.

Usage Guidelines

The inspect h323 command provides support for H.323 compliant applications such as Cisco CallManager and VocalTec Gatekeeper. H.323 is a suite of protocols defined by the International Telecommunication Union (ITU) for multimedia conferences over LANs. The ASA supports H.323 through Version 6, including the H.323 v3 feature Multiple Calls on One Call Signaling Channel.

With H.323 inspection enabled, the ASA supports multiple calls on the same call signaling channel, a feature added with H.323 Version 3. This feature reduces call setup time and reduces the use of ports on the ASA.

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, the ASA uses an ASN.1 decoder to decode the H.323 messages.
- Dynamically allocate the negotiated H.245 and RTP/RTCP connections.

Inspecting Signaling Messages

For inspecting signaling messages, the **inspect h323** command often needs to determine locations of the media endpoints (for example, IP phones).

This information is used to prepare access control and NAT state for media traffic to traverse the firewall transparently without manual configuration.

In determining these locations, the **inspect h323** command does **not** use the tunnel default gateway route. A tunnel default gateway route is a route of the form **route** *interface* **0 0** *metric* **tunneled**. This route overrides the default route for packets that egress from IPsec tunnels. Therefore, if the **inspect h323** command is desired for VPN traffic, do not configure the tunnel default gateway route. Instead, use other static routing or dynamic routing.

Examples

The following example enables the H.323 inspection engine, which creates a class map to match H.323 traffic on the default port (1720). The service policy is then applied to the outside interface.

```
ciscoasa(config)# class-map h323-port
ciscoasa(config-cmap)# match port tcp eq 1720
ciscoasa(config-cmap)# exit
ciscoasa(config)# policy-map h323_policy

ciscoasa(config-pmap)# class h323-port
ciscoasa(config-pmap-c)# inspect h323
ciscoasa(config-pmap-c)# exit
ciscoasa(config)# service-policy h323_policy interface outside
```

Related Commands

Commands	Description
policy-map type inspect	Creates an inspection policy map.
show h225	Displays information for H.225 sessions established across the ASA.
show h245	Displays information for H.245 sessions established across the ASA by endpoints using slow start.
show h323 ras	Displays information for H.323 RAS sessions established across the ASA.
timeout {**h225** | **h323**}	Configures idle time after which an H.225 signaling connection or an H.323 control connection will be closed.

inspect http

To enable HTTP application inspection or to change the ports to which the ASA listens, use the **inspect http command** in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect http [*map_name*]
no inspect http [*map_name*]

Syntax Description

map_name (Optional) The name of the HTTP inspection map.

Command Default

The default port for HTTP is 80.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added. It replaced the **fixup** command, which has been deprecated.

Usage Guidelines

Tip You can install a service module that performs application and URL filtering, which includes HTTP inspection, such as ASA CX or ASA FirePOWER. The HTTP inspection running on the ASA is not compatible with these modules. Note that it is far easier to configure application filtering using a purpose-built module rather than trying to manually configure it on the ASA using an HTTP inspection policy map.

Use the HTTP inspection engine to protect against specific attacks and other threats that are associated with HTTP traffic.

HTTP application inspection scans HTTP headers and body, and performs various checks on the data. These checks prevent various HTTP constructs, content types, and tunneling and messaging protocols from traversing the security appliance.

The enhanced HTTP inspection feature, which is also known as an application firewall and is available when you configure an HTTP inspection policy map, can help prevent attackers from using HTTP messages for circumventing network security policy.

HTTP application inspection can block tunneled applications and non-ASCII characters in HTTP requests and responses, preventing malicious content from reaching the web server. Size limiting of various elements in HTTP request and response headers, URL blocking, and HTTP server header type spoofing are also supported.

Enhanced HTTP inspection verifies the following for all HTTP messages:

- Conformance to RFC 2616
- Use of RFC-defined methods only.
- Compliance with the additional criteria.

Examples

In this example, any HTTP connection (TCP traffic on port 80) that enters the ASA through any interface is classified for HTTP inspection. Because the policy is a global policy, inspection occurs only as the traffic enters each interface.

```
ciscoasa(config)# class-map http_traffic
ciscoasa(config-cmap)# match port tcp eq 80
ciscoasa(config)# policy-map http_traffic_policy
ciscoasa(config-pmap)# class http_traffic
ciscoasa(config-pmap-c)# inspect http
ciscoasa(config)# service-policy http_traffic_policy global
```

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
policy-map	Associates a class map with specific security actions.
policy-map type inspect	Creates an inspection policy map.

inspect icmp

To configure the ICMP inspection engine, use the **inspect icmp** command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect icmp
no inspect icmp

Command Default

This command is disabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added. It replaced the **fixup** command, which has been deprecated.

Usage Guidelines

The ICMP inspection engine allows ICMP traffic to be inspected like TCP and UDP traffic. Without the ICMP inspection engine, we recommend that you do not allow ICMP through the ASA 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

When ICMP inspection is disabled, which is the default configuration, ICMP echo reply messages are denied from a lower security interface to a higher security interface, even if it is in response to an ICMP echo request.

Examples

You enable the ICMP application inspection engine as shown in the following example, which creates a class map to match ICMP traffic using the ICMP protocol ID, which is 1 for IPv4 and 58 for IPv6. The service policy is then applied to the outside interface. To enable ICMP inspection for all interfaces, use the **global** parameter in place of **interface outside**.

```
ciscoasa(config)# class-map icmp-class
ciscoasa(config-cmap)# match default-inspection-traffic
ciscoasa(config-cmap)# exit
ciscoasa(config)# policy-map icmp_policy

ciscoasa(config-pmap)# class icmp-class
ciscoasa(config-pmap-c)# inspect icmp
ciscoasa(config-pmap-c)# exit
ciscoasa(config)# service-policy icmp_policy interface outside
```

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
icmp	Configures access rules for ICMP traffic that terminates at an ASA interface.
policy-map	Defines a policy that associates security actions with one or more traffic classes.
service-policy	Applies a policy map to one or more interfaces.

inspect icmp error

To enable application inspection for ICMP error messages, use the **inspect icmp** error command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect icmp error
no inspect icmp error

Command Default

This command is disabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added. It replaced the **fixup** command, which has been deprecated.

Usage Guidelines

When ICMP Error inspection is enabled, the ASA creates translation sessions for intermediate hops that send ICMP error messages, based on the NAT configuration. The ASA overwrites the packet with the translated IP addresses.

When disabled, the ASA 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 ASA 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 ASA. When the ASA does not translate the intermediate hops, all the intermediate hops appear with the mapped destination IP address.

The ICMP payload is scanned to retrieve the five-tuple from the original packet. Using the retrieved five-tuple, a lookup is performed to determine the original address of the client. The ICMP error inspection engine makes the following changes to the ICMP packet:

- In the IP Header, the mapped IP is changed to the real 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 mapped IP is changed to the real IP
  - Original packet mapped port is changed to the real Port

- Original packet IP checksum is recalculated

Examples

The following example enables the ICMP error application inspection engine, which creates a class map to match ICMP traffic using the ICMP protocol ID, which is 1 for IPv4 and 58 for IPv6. The service policy is then applied to the outside interface. To enable ICMP error inspection for all interfaces, use the **global** parameter in place of **interface outside**.

```
ciscoasa(config)# class-map icmp-class
ciscoasa(config-cmap)# match default-inspection-traffic
ciscoasa(config-cmap)# exit
ciscoasa(config)# policy-map icmp_policy

ciscoasa(config-pmap)# class icmp-class
ciscoasa(config-pmap-c)# inspect icmp error
ciscoasa(config-pmap-c)# exit
ciscoasa(config)# service-policy icmp_policy interface outside
```

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
icmp	Configures access rules for ICMP traffic that terminates at an ASA interface.
inspect icmp	Enables or disables the ICMP inspection engine.
policy-map	Defines a policy that associates security actions with one or more traffic classes.
service-policy	Applies a policy map to one or more interfaces.

inspect ils

To enable ILS application inspection, use the inspect **ils command** in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect ils
no inspect ils

Command Default

This command is disabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added. It replaced the **fixup** command, which has been deprecated.

Usage Guidelines

The **inspect ils** command provides NAT support for Microsoft NetMeeting, SiteServer, and Active Directory products that use LDAP to exchange directory information with an ILS server.

The ASA supports NAT 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 are searched first, and then DNAT entries to obtain the correct address. If both of these searches fail, then the address is not changed. For sites using NAT 0 (no NAT) and not expecting DNAT interaction, we recommend that the inspection engine be turned off to provide better performance.

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

Because ILS traffic only occurs on the secondary UDP channel, the TCP connection is disconnected after the TCP inactivity interval. By default, this interval is 60 minutes and can be adjusted using the **timeout** command.

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.

Note Because H.225 call signaling traffic only occurs on the secondary UDP channel, the TCP connection is disconnected after the interval specified by the TCP **timeout** command. By default, this interval is set at 60 minutes.

Examples

You enable the ILS inspection engine as shown in the following example, which creates a class map to match ILS traffic on the default port (389). The service policy is then applied to the outside interface. To enable ILS inspection for all interfaces, use the **global** parameter in place of **interface outside**.

```
ciscoasa(config)# class-map ils-port
ciscoasa(config-cmap)# match port tcp eq 389
ciscoasa(config-cmap)# exit
ciscoasa(config)# policy-map ils_policy

ciscoasa(config-pmap)# class ils-port
ciscoasa(config-pmap-c)# inspect ils
ciscoasa(config-pmap-c)# exit
ciscoasa(config)# service-policy ils_policy interface outside
```

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
policy-map	Associates a class map with specific security actions.

Commands	Description
policy-map type inspect	Creates an inspection policy map.
service-policy	Applies a policy map to one or more interfaces.

inspect im

To enable inspection of Instant Messenger traffic, use the inspect im command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect im *map_name*
no inspect im *map_name*

Syntax Description

map_name	The name of the IM map.

Command Default

This command is disabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.2(1)	This command was added.

Usage Guidelines

The **inspect im** command enables or disables application inspection for the IM protocol. The Instant Messaging (IM) inspect engine lets you control the network usage of IM and stop leakage of confidential data, propagation of worms, and other threats to the corporate network.

Examples

The following example shows how to define an IM inspection policy map:

```
ciscoasa(config)# regex loginname1 "user1\@example.com"
ciscoasa(config)# regex loginname2 "user2\@example.com"
ciscoasa(config)# regex loginname3 "user3\@example.com"
ciscoasa(config)# regex loginname4 "user4\@example.com"
ciscoasa(config)# regex yahoo_version_regex "1\.0"
ciscoasa(config)# regex gif_files ".*\.gif"
ciscoasa(config)# regex exe_files ".*\.exe"
ciscoasa(config)# class-map type regex match-any yahoo_src_login_name_regex
ciscoasa(config-cmap)# match regex loginname1
ciscoasa(config-cmap)# match regex loginname2
ciscoasa(config)# class-map type regex match-any yahoo_dst_login_name_regex
ciscoasa(config-cmap)# match regex loginname3
ciscoasa(config-cmap)# match regex loginname4
ciscoasa(config)# class-map type inspect im match-any yahoo_file_block_list
ciscoasa(config-cmap)# match filename regex gif_files
```

```
ciscoasa(config-cmap)# match filename regex exe_files

ciscoasa(config)# class-map type inspect im match-all yahoo_im_policy
ciscoasa(config-cmap)# match login-name regex class yahoo_src_login_name_regex
ciscoasa(config-cmap)# match peer-login-name regex class yahoo_dst_login_name_regex
ciscoasa(config)# class-map type inspect im match-all yahoo_im_policy2
ciscoasa(config-cmap)# match version regex yahoo_version_regex
ciscoasa(config)# class-map im_inspect_class_map
ciscoasa(config-cmap)# match default-inspection-traffic
ciscoasa(config)# policy-map type inspect im im_policy_all
ciscoasa(config-pmap)# class yahoo_file_block_list
ciscoasa(config-pmap-c)# match service file-transfer
ciscoasa(config-pmap)# class yahoo_im_policy
ciscoasa(config-pmap-c)# drop-connection
ciscoasa(config-pmap)# class yahoo_im_policy2
ciscoasa(config-pmap-c)# reset
ciscoasa(config)# policy-map global_policy_name
ciscoasa(config-pmap)# class im_inspect_class_map
ciscoasa(config-pmap-c)# inspect im im_policy_all
```

Related Commands

Commands	Description
class	Identifies a class map name in the policy map.
policy-map	Creates a Layer 3/4 policy map.
policy-map type inspect	Creates an inspection policy map.
show running-config policy-map	Display all current policy map configurations.
match protocol	Matches a specific IM protocol in an inspection class or policy map.

inspect ip-options

To enable inspection of IP options in a packet header, use the inspect ip-options command in class or policy map type inspect configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect ip-options [*map_name*]
no inspect ip-options *map_name*

Syntax Description

map_name	(Optional.) The name of the IP Options map.

Command Default

This command is enabled by default in the global policy. The default inspection map allows packets with the router-alert option, but drops packets that have any other options.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Policy or class map configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
8.2(2)	This command was added. Supported options are **eool**, **nop**, and **router-alert** options. If an IP header contains additional options other than EOOL, NOP, or RTRALT, regardless of whether the ASA is configured to allow these options, the ASA will drop the packet.
9.5(1)	Support for all IP options was added.

Usage Guidelines

In a packet, the IP header contains the Options field. The Options field, commonly referred to as IP Options, provides for control functions that are required in some situations but unnecessary for most common communications. In particular, IP Options include provisions for time stamps, security, and special routing. Use of IP Options is optional and the field can contain zero, one, or more options.

You can configure IP Options inspection to control which IP packets are allowed based on the contents of the IP Options field in the packet header. You can drop packets that have unwanted options, clear the options (and allow the packet), or allow the packet without change.

If you want non-default processing, create an IP Options inspection policy map, enter the **parameter** command, and specify the actions to take for the various options. You can inspect the following options. In all cases, the allow action allows packets that contain the option without modification; the clear action allows the packets but removes the option from the header.

Use the **no** form of the command to remove the option from the map. Any packet that contains an option that you do not include in the map is dropped, even if the packet contains otherwise allowed or cleared options.

For a list of IP options, with references to the relevant RFCs, see the IANA page, http://www.iana.org/assignments/ip-parameters/ip-parameters.xhtml .

- **default action** {**allow** | **clear**}—Sets the default action for any option not explicitly included in the map. If you do not set a default action of allow or clear, packets that contain non-allowed options are dropped.
- **basic-security action** {**allow** | **clear**}—Allows or clears the Security (SEC) option.
- **commercial-security action** {**allow** | **clear**}—Allows or clears the Commercial Security (CIPSO) option.
- **eool action** {**allow** | **clear**}—Allows or clears the End of Options List option. This option, which contains just a single zero byte, appears at the end of all options to mark the end of a list of options. This might not coincide with the end of the header according to the header length.
- **exp-flow-control action** {**allow** | **clear**}—Allows or clears the Experimental Flow Control (FINN) option.
- **exp-measurement action** {**allow** | **clear**}—Allows or clears the Experimental Measurement (ZSU) option.
- **extended-security action** {**allow** | **clear**}—Allows or clears the Extended Security (E-SEC) option.
- **imi-traffic-descriptor action** {**allow** | **clear**}—Allows or clears the IMI Traffic Descriptor (IMITD) option.
- **nop action** {**allow** | **clear**}—Allows or clears the No Operation option. The Options field in the IP header can contain zero, one, or more options, which makes the total length of the field variable. However, the IP header must be a multiple of 32 bits. If the number of bits of all options is not a multiple of 32 bits, the NOP option is used as “internal padding” to align the options on a 32-bit boundary.
- **quick-start action** {**allow** | **clear**}—Allows or clears the Quick-Start (QS) option.
- **record-route action** {**allow** | **clear**}—Allows or clears the Record Route (RR) option.
- **router-alert action** {**allow** | **clear**}.—Allows or clears the Router Alert (RTRALT) option. This option is allowed in the default IP Options inspection policy map. This option notifies transit routers to inspect the contents of the packet even when the packet is not destined for that router. This inspection is valuable when implementing RSVP and similar protocols that require relatively complex processing from the routers along the packets delivery path. Dropping RSVP packets containing the Router Alert option can cause problems in VoIP implementations.
- **timestamp action** {**allow** | **clear**}—Allows or clears the Time Stamp (TS) option.
- {**0-255**} **action** {**allow** | **clear**}—Allows or clears the option identified by the option type number. The number is the whole option type octet (copy, class, and option number), not just the option number portion of the octet. These option types might not represent real options. Non-standard options must be in the expected type-length-value format defined in the Internet Protocol RFC 791, http://tools.ietf.org/html/rfc791 .

Examples

The following example shows how to define an IP Options inspection policy map that allows the ASA to pass packets that contain the EOOL, NOP, and RTRALT options in the packet header.

```
ciscoasa(config)# policy-map type inspect ip-options ip-options-map
```

```
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)# eool action allow

ciscoasa(config-pmap-p)# nop action allow

ciscoasa(config-pmap-p)# router-alert action allow
```

The following example shows how to set a new default action to allow packets with any IP options.

```
ciscoasa(config)# policy-map type inspect ip-options ip-options-map
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)# default action allow
```

Related Commands

Commands	Description
class	Identifies a class map name in the policy map.
policy-map	Creates a Layer 3/4 policy map.
policy-map type inspect	Creates an inspection policy map.

inspect ipsec-pass-thru

To enable IPsec pass-through inspection, use the inspect ipsec-pass-thru command in class map configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect ipsec-pass-thru [*map_name*]
no inspect ipsec-pass-thru [*map_name*]

Syntax Description

map_name	(Optional) The name of the IPsec pass-through map.

Command Default

This command is disabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added.

Usage Guidelines

The **inspect ipsec-pass-thru** command enables or disables application inspection. IPsec pass-through application inspection provides convenient traversal of ESP (IP protocol 50) and/or AH (IP protocol 51) traffic associated with an IKE UDP port 500 connection. It avoids lengthy access list configuration to permit ESP and AH traffic and also provides security using timeout and maximum connections.

Use the IPsec pass-through parameter map to identify a specific map to use for defining the parameters for the inspection. Use the policy-map type inspect command to access the parameters configuration, which lets you specify the restrictions for ESP or AH traffic. You can set the per-client maximum connections and the idle timeout in parameters configuration mode.

Use the class-map, policy-map, and service-policy commands to define a class of traffic, to apply the inspect command to the class, and to apply the policy to one or more interfaces. The parameter map defined is enabled when used with the inspect ipsec-pass-thru command.

NAT and non-NAT traffic is permitted. However, PAT is not supported.

Note In ASA 7.0(1), the **inspect ipsec-pass-thru** command allowed only ESP traffic to pass through. To retain the same behavior in later versions, a default map that permits ESP is created and attached if the **inspect ipsec-pass-thru** command is specified without any arguments. This map can be seen in the output of the show running-config all command.

Examples

The following example shows how to use access lists to identify IKE traffic, define an IPsec pass-through parameter map, define a policy, and apply the policy to the outside interface:

```
ciscoasa(config)# access-list ipsecpassthruacl permit udp any any eq 500
ciscoasa(config)# class-map ipsecpassthru-traffic
ciscoasa(config-cmap)# match access-list ipsecpassthruacl
ciscoasa(config)# policy-map type inspect ipsec-pass-thru iptmap
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)# esp per-client-max 10 timeout 0:11:00
ciscoasa(config-pmap-p)# ah per-client-max 5 timeout 0:06:00
ciscoasa(config)# policy-map inspection_policy
ciscoasa(config-pmap)# class ipsecpassthru-traffic
ciscoasa(config-pmap-c)# inspect ipsec-pass-thru iptmap
ciscoasa(config)# service-policy inspection_policy interface outside
```

Related Commands

Commands	Description
class	Identifies a class map name in the policy map.
class-map type inspect	Creates an inspection class map to match traffic specific to an application.
policy-map	Creates a Layer 3/4 policy map.
show running-config policy-map	Display all current policy map configurations.
match protocol	Matches a specific IM protocol in an inspection class or policy map.

inspect ipv6

To enable IPv6 inspection, use the inspect ipv6 command in class configuration mode. Class configuration mode is accessible from policy-map configuration mode. To remove the configuration, use the **no** form of this command.

inspect ipv6 [*map_name*]
no inspect ipv6 [*map_name*]

Syntax Description

map_name	(Optional.) The name of the IPv6 inspection policy map.

Command Default

IPv6 inspection is disabled by default.

If you enable IPv6 inspection and do not specify an inspection policy map, then the default IPv6 inspection policy map is used, and the following actions are taken:

- Allows only known IPv6 extension headers. Non-conforming packets are dropped and logged.
- Enforces the order of IPv6 extension headers as defined in the RFC 2460 specification. Non-conforming packets are dropped and logged.
- Drops any packet with a routing type header.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
8.2(1)	This command was added.

Usage Guidelines

IPv6 inspection lets you selectively log or drop IPv6 traffic based on the extension header. In addition, IPv6 inspection can check conformance to RFC 2460 for type and order of extension headers in IPv6 packets.

Examples

The following example drops all IPv6 traffic with the hop-by-hop, destination-option, routing-address, and routing type 0 headers:

```
policy-map type inspect ipv6 ipv6-pm
 parameters
 match header hop-by-hop
  drop
 match header destination-option
```

```
  drop
 match header routing-address count gt 0
  drop
 match header routing-type eq 0
  drop
policy-map global_policy
 class class-default
  inspect ipv6 ipv6-pm
!
service-policy global_policy global
```

Related Commands

Commands	Description
class	Identifies a class map name in the policy map.
match header	Matches IPv6 headers in an IPv6 inspection policy map.
policy-map type inspect ipv6	Creates an inspection policy map for IPv6.
policy-map	Creates a Layer 3/4 policy map.
verify-header	Configures IPv6 inspection parameters.

inspect lisp

To enable LISP inspection, use the **inspect lisp** command in class configuration mode. You can access the class configuration mode by first entering the **policy-map** command. To disable LISP inspection, use the **no** form of this command.

inspect lisp [*inspect_map_name*]
no inspect lisp [*inspect_map_name*]

Syntax Description

inspect_map_name	Specify the LISP inspection map name (**policy-map type inspect lisp**) if you want to limit the EIDs or if you need to specify the pre-shared key for LISP messages.

Command Default

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
9.5(2)	We added this command.

Usage Guidelines

The ASA inspects LISP traffic for the EID-notify message sent between the first hop router and the ITR or ETR. The ASA maintains an EID table that correlates the EID and the site ID.

About LISP Inspection for Cluster Flow Mobility

The ASA inspects LISP traffic for location changes and then uses this information for seamless clustering operation. With LISP integration, the ASA cluster members can inspect LISP traffic passing between the first hop router and the ETR or ITR, and can then change the flow owner to be at the new site.

Cluster flow mobility includes several inter-related configurations:

1. (Optional) Limit inspected EIDs based on the host or server IP address—The first hop router might send EID-notify messages for hosts or networks the ASA cluster is not involved with, so you can limit the EIDs to only those servers or networks relevant to your cluster. For example, if the cluster is only involved with 2 sites, but LISP is running on 3 sites, you should only include EIDs for the 2 sites involved with the cluster. See the **policy-map type inspect lisp**, **allowed-eid,** and **validate-key** commands.
2. LISP traffic inspection—The ASA inspects LISP traffic for the EID-notify message sent between the first hop router and the ITR or ETR. The ASA maintains an EID table that correlates the EID and the site ID. For example, you should inspect LISP traffic with a source IP address of the first hop router and a destination address of the ITR or ETR. See the **inspect lisp** command.

3. Service Policy to enable flow mobility on specified traffic—You should enable flow mobility on business-critical traffic. For example, you can limit flow mobility to only HTTPS traffic, and/or to traffic to specific servers. See the **cluster flow-mobility lisp** command.
4. Site IDs—The ASA uses the site ID for each cluster unit to determine the new owner. See the **site-id** command.
5. Cluster-level configuration to enable flow mobility—You must also enable flow mobility at the cluster level. This on/off toggle lets you easily enable or disable flow mobility for a particular class of traffic or applications. See the **flow-mobility lisp** command.

Examples

The following example Inspects LISP traffic (UDP 4342) between a LISP router at 192.168.50.89 (on inside) and an ITR or ETR router (on another ASA interface) at 192.168.10.8:

```
ciscoasa(config)# access-list LISP_ACL extended permit udp host 192.168.50.89 host
192.168.10.8 eq 4342
ciscoasa(config)# class-map LISP_CLASS
ciscoasa(config-cmap)# match access-list LISP_ACL
ciscoasa(config-cmap)# policy-map INSIDE_POLICY
ciscoasa(config-pmap)# class LISP_CLASS
ciscoasa(config-pmap-c)# inspect lisp LISP_EID_INSPECT
ciscoasa(config)# service-policy INSIDE_POLICY interface inside
```

Related Commands

Command	Description
allowed-eids	Limits inspected EIDs based on IP address.
clear cluster info flow-mobility counters	Clears the flow mobility counters.
clear lisp eid	Removes EIDs from the ASA EID table.
cluster flow-mobility lisp	Enables flow mobility for the service policy.
flow-mobility lisp	Enables flow mobility for the cluster.
inspect lisp	Inspects LISP traffic.
policy-map type inspect lisp	Customizes the LISP inspection.
site-id	Sets the site ID for a cluster chassis.
show asp table classify domain inspect-lisp	Shows the ASP table for LISP inspection.
show cluster info flow-mobility counters	Shows flow mobility counters.
show conn	Shows traffic subject to LISP flow-mobility.
show lisp eid	Shows the ASA EID table.
show service-policy	Shows the service policy.
validate-key	Enters the pre-shared key to validate LISP messages.

inspect m3ua

To enable M3UA inspection, use the **inspect m3ua** command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. Use the **no** form of this command to disable M3UA inspection.

inspect m3ua [*map_name*]
no inspect m3ua [*map_name*]

Note M3UA inspection requires the Carrier license.

Syntax Description

map_name (Optional) Name for the M3UA inspection policy map.

Command Default

This command is disabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
9.6(2)	This command was added.

Usage Guidelines

MTP3 User Adaptation (M3UA) is a client/server protocol that provides a gateway to the SS7 network for IP-based applications that interface with the SS7 Message Transfer Part 3 (MTP3) layer. M3UA makes it possible to run the SS7 User Parts (such as ISUP) over an IP network. M3UA is defined in RFC 4666.

M3UA uses SCTP as the transport layer. SCTP port 2905 is the expected port, although you can configure the signaling gateways to use a different port.

The MTP3 layer provides networking functions such as routing and node addressing, but uses point codes to identify nodes. The M3UA layer exchanges Originating Point Codes (OPC) and Destination Point Codes (DPC). This is similar to how IP uses IP addresses to identify nodes.

M3UA inspection provides limited protocol conformance.

You can optionally create an M3UA inspection policy map to apply access policy based on point codes or Service Indicators (SI). You can also apply rate limiting based on message class and type.

Examples

The following example shows an M3UA inspection policy map and inspection policy.

```
ciscoasa(config)# policy-map type inspect m3ua m3ua-map
ciscoasahostname(config-pmap-c)# drop
ciscoasa(config-pmap-c)# match message class 9
ciscoasa(config-pmap-c)# drop
ciscoasa(config-pmap-c)# match dpc 1-5-1
ciscoasa(config-pmap-c)# drop log
ciscoasa(config-pmap-c)# parameters
ciscoasa(config-pmap-p)# ss7 variant ITU
ciscoasa(config-pmap-p)# timeout endpoint 00:45:00
ciscoasa(config)# policy-map global_policy
ciscoasa(config-pmap)# class inspection_default
ciscoasa(config-pmap-c)# inspect m3ua m3ua-map
ciscoasa(config)# service-policy global_policy global
```

Related Commands

Commands	Description
class	Defines the traffic class to which to apply security actions.
policy-map type inspect	Creates an inspection policy map.
service-policy	Applies a policy map to one or more interfaces.
show service-policy inspect m3ua	Shows that status and statistics of the inspect m3ua policy.

inspect mgcp

To enable MGCP application inspection or to change the ports to which the ASA listens, use the inspect **mgcp** command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect mgcp [*map_name*]
no inspect mgcp [*map_name*]

Syntax Description

map_name	(Optional) The name of the MGCP map.

Command Default

This command is disabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added. It replaced the **fixup** command, which has been deprecated.

Usage Guidelines

To use MGCP, you usually need to configure at least two **inspect** commands: one for the port on which the gateway receives commands, and one for the port on which the Call Agent receives commands. Normally, a Call Agent sends commands to the default MGCP port for gateways, 2427, and a gateway sends commands to the default MGCP port for Call Agents, 2727.

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. Using NAT and PAT with MGCP lets you support a large number of devices on an internal network with a limited set of external (global) addresses.

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, and 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.

Note MGCP call agents send AUEP messages to determine if MGCP end points are present. This establishes a flow through the ASA and allows MGCP end points to register with the call agent.

Use the **call-agent** and **gateway** commands in MGCP map configuration mode to configure the IP addresses of one or more call agents and gateways. Use the **command-queue** command in MGCP map configuration mode to specify the maximum number of MGCP commands that will be allowed in the command queue at one time.

Inspecting Signaling Messages

For inspecting signaling messages, the **inspect mgcp** command often needs to determine locations of the media endpoints (for example, IP phones).

This information is used to prepare access-control and NAT state for media traffic to traverse the firewall transparently without manual configuration.

In determining these locations, the **inspect mgcp** command does **not** use the tunnel default gateway route. A tunnel default gateway route is a route of the form **route** *interface* **0 0** *metric* **tunneled**. This route overrides the default route for packets that egress from IPsec tunnels. Therefore, if the **inspect mgcp** command is desired for VPN traffic, do not configure the tunnel default gateway route. Instead, use other static routing or dynamic routing.

The maximum number of MGCP commands that can be queued is 150.

Examples

The following example shows how to identify MGCP traffic, define a MGCP inspection map, define a policy, and apply the policy to the outside interface. This creates a class map to match MGCP traffic on the default ports (2427 and 2727). The service policy is then applied to the outside interface. This configuration 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 both gateways 10.10.10.116 and 10.10.10.117. To enable MGCP inspection for all interfaces, use the **global** parameter in place of **interface outside**.

```
ciscoasa(config)# access-list mgcp_acl permit tcp any any eq 2427

ciscoasa(config)# access-list mgcp_acl permit tcp any any eq 2727
ciscoasa(config)# class-map mgcp_port
ciscoasa(config-cmap)# match access-list mgcp_acl
ciscoasa(config-cmap)# exit
ciscoasa(config)# policy-map type inspect mgcp inbound_mgcp
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)# call-agent 10.10.11.5 101
ciscoasa(config-pmap-p)# call-agent 10.10.11.6 101
ciscoasa(config-pmap-p)# call-agent 10.10.11.7 102
ciscoasa(config-pmap-p)# call-agent 10.10.11.8 102
ciscoasa(config-pmap-p)# gateway 10.10.10.115 101
ciscoasa(config-pmap-p)# gateway 10.10.10.116 102
ciscoasa(config-pmap-p)# gateway 10.10.10.117 102
ciscoasa(config-pmap-p)# command-queue 150
ciscoasa(config-mgcp-map)# exit
```

```
ciscoasa(config)# policy-map inbound_policy
ciscoasa(config-pmap)# class mgcp_port
ciscoasa(config-pmap-c)# inspect mgcp
mgcp-map inbound_mgcp
ciscoasa(config-pmap-c)# exit
ciscoasa(config)# service-policy inbound_policy interface outside
```

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
policy-map type inspect mgcp	Creates an inspection policy map for MGCP.
show mgcp	Displays information about MGCP sessions established through the ASA.
timeout	Sets the maximum idle time duration for different protocols and session types.

inspect mmp

To configure the MMP inspection engine, use the **inspect mmp** command in class configuration mode. To remove MMP inspection, use the **no** form of this command.

inspect mmp tls-proxy [*name*]
no inspect mmp tls-proxy [*name*]

Syntax Description

name	Species the TLS proxy instance name.
tls-proxy	Enables the TLS proxy for MMP inspection. The MMP protocol can additionally use the TCP transport; however, the CUMA client only supports the TLS transport. Therefore, the **tls-proxy** keyword is required to enable MMP inspection.

Command Default

This command is disabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
8.0(4)	The command was added.

Usage Guidelines

The ASA includes an inspection engine to validate the CUMA Mobile Multiplexing Protocol (MMP). MMP is a data transport protocol for transmitting data entities between CUMA clients and servers. Use the **inspect mmp** command when the ASA is deployed between CUMA clients and servers and inspection of MMP packets is required.

MMP inspection must be enabled with the TLS proxy because MMP traffic is transported only over a TLS connection.

Note While configuring the MMP inspection engine, please note that it can only be added under a non-default inspection class.

Examples

The following example shows the use of the **inspect mmp** command to inspect MMP traffic:

```
ciscoasa
```

```
(config)#
class-map mmp
ciscoasa
(config-cmap)#
match port tcp eq 5443
ciscoasa
(config-cmap)#
exit
ciscoasa
(config)#
policy-map mmp-policy
ciscoasa
(config-pmap)#
class mmp
ciscoasa(config-pmap-c)# inspect mmp tls-proxy myproxy
ciscoasa(config-pmap-c)# exit
ciscoasa(config-pmap)# exit
ciscoasa
(config)#
service-policy mmp-policy interface outside
```

Related Commands

Command	Description
tls-proxy	Configures the TLS proxy instance.

inspect netbios

To enable NetBIOS application inspection or to change the ports to which the ASA listens, use the inspect netbios **command** in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect netbios [*map_name*]
no inspect netbios [*map_name*]

Syntax Description

map_name	(Optional) The name of the NetBIOS map.

Command Default

This command is enabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added. It replaced the **fixup** command, which has been deprecated.

Usage Guidelines

The **inspect netbios** command enables or disables application inspection for the NetBIOS protocol. NetBIOS inspection is enabled by default. The NetBIOS inspection engine translates IP addresses in the NetBIOS name service (NBNS) packets according to the ASA NAT configuration. You can optionally create a policy map to drop or log NetBIOS protocol violations.

Examples

The following example shows how to define a NetBIOS inspection policy map:

```
ciscoasa(config)# policy-map type inspect netbios netbios_map
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)# protocol-violation drop
```

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
policy-map	Associates a class map with specific security actions.
policy-map type inspect netbios	Creates an inspection policy map for NetBIOS.

Commands	Description
service-policy	Applies a policy map to one or more interfaces.

inspect pptp

To enable PPTP application inspection or to change the ports to which the ASA listens, use the inspect **pptp** command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect pptp
no inspect pptp

Syntax Description

This command has no arguments or keywords.

Command Default

This command is disabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added. It replaced the **fixup** command, which has been deprecated.

Usage Guidelines

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.

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].

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

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

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.

To enable PPTP inspection for all interfaces, use the **global** parameter in place of **interface outside**.

Examples

You enable the PPTP inspection engine as shown in the following example, which creates a class map to match PPTP traffic on the default port (1723). The service policy is then applied to the outside interface.

```
ciscoasa(config)# class-map pptp-port
ciscoasa(config-cmap)# match port tcp eq 1723
ciscoasa(config-cmap)# exit
ciscoasa(config)# policy-map pptp_policy
ciscoasa(config-pmap)# class pptp-port
ciscoasa(config-pmap-c)# inspect pptp
ciscoasa(config-pmap-c)# exit
ciscoasa(config)# service-policy pptp_policy interface outside
```

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
policy-map	Associates a class map with specific security actions.
service-policy	Applies a policy map to one or more interfaces.

inspect radius-accounting

To enable or disable RADIUS accounting inspection or to define a map for controlling traffic or tunnels, use the **inspect radius-accounting** command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect radius-accounting *map_name*
no inspect radius-accounting [*map_name*]

Syntax Description

map_name	Name for the RADIUS accounting map.

Command Default

This command is disabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.2(1)	This command was added.

Usage Guidelines

The purpose of RADIUS accounting inspection is to prevent over-billing attacks on GPRS networks that use RADIUS servers. Although you do not need the GTP/GPRS or Carrier license to implement RADIUS accounting inspection, it has no purpose unless you are implementing GTP inspection and you have a GPRS setup.

Use the **policy-map type inspect radius-accounting** command to create an inspection map to use for defining the parameters for RADIUS accounting. After entering the parameters command, you can define the inspection characteristics and behavior using the **send response**, **host**, **validate-attribute**, **enable gprs**, and **timeout users** commands.

Then you use the **class-map type management**, **policy-map**, and **service-policy** commands to define a class of traffic, to apply the inspect radius-accounting command to the class, and to apply the policy to one or more interfaces.

Note The **inspect radius-accounting** command can only be used with the **class-map type management** command.

Examples

The following example shows how to configure a RADIUS accounting inspection map and enable inspection globally.

```
policy-map type inspect radius-accounting radius-acct-pmap
  parameters
    send response
    enable gprs
    validate-attribute 31
    host 10.2.2.2 key 123456789
    host 10.1.1.1 key 12345
class-map type management radius-class
  match port udp eq radius-acct
policy-map global_policy
  class radius-class
    inspect radius-accounting radius-acct-pmap
```

Related Commands

Commands	Description
parameters	Defines the traffic class to which to apply security actions.
class-map type management	Lets you identify Layer 3 or 4 management traffic destined for the ASA to which you want to apply actions.
policy-map type inspect radius-accounting	Creates an inspection policy map for RADIUS accounting.
show and **clear service-policy**	Lets you view and clear service policy settings.
service-policy	Applies a policy map to one or more interfaces.

inspect rsh

To enable RSH application inspection or to change the ports to which the ASA listens, use the inspect **rsh** command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect rsh
no inspect rsh

Syntax Description

This command has no arguments or keywords.

Command Default

This command is enabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added. It replaced the **fixup** command, which has been deprecated.

Usage Guidelines

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 listens for the STDERR output stream. RSH inspection supports NAT of the negotiated port number if necessary.

Examples

The following example enables the RSH inspection engine, which creates a class map to match RSH traffic on the default port (514). The service policy is then applied to the outside interface. To enable RSH inspection for all interfaces, use the **global** parameter in place of **interface outside**.

```
ciscoasa(config)# class-map rsh-port
ciscoasa(config-cmap)# match port tcp eq 514
ciscoasa(config-cmap)# exit
ciscoasa(config)# policy-map rsh_policy

ciscoasa(config-pmap)# class rsh-port
ciscoasa(config-pmap-c)# inspect rsh
ciscoasa(config-pmap-c)# exit
ciscoasa(config)# service-policy rsh_policy interface outside
```

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
policy-map	Associates a class map with specific security actions.
service-policy	Applies a policy map to one or more interfaces.

inspect rtsp

To enable RTSP application inspection or to change the ports to which the ASA listens, use the inspect **rtsp** command in class configuration mode. Class configuration mode is accessible from policy-map configuration mode. To remove the configuration, use the **no** form of this command.

inspect rtsp [*map_name*]
no inspect rtsp [*map_name*]

Syntax Description

map_name	(Optional) The name of the RTSP map.

Command Default

This command is enabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added. It replaced the **fixup** command, which has been deprecated.

Usage Guidelines

The **inspect rtsp** command lets the ASA pass RTSP packets. RTSP is used by RealAudio, RealNetworks, Apple QuickTime 4, RealPlayer, and Cisco IP/TV connections.

Note For Cisco IP/TV, use RTSP TCP port 554 and TCP 8554.

RTSP applications use the well-known port 554 with TCP (rarely UDP) as a control channel.

The ASA only supports TCP, in conformity with RFC 2326. This TCP control channel is 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.

Note The inspection queue is typically used for handling control packets only. Unnecessary forwarding of media packets to the inspection queue can cause queue or block exhaustion. Hence, you must enable **inspect rtsp** with caution.

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

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

Using RealPlayer

When using RealPlayer, it is important to properly configure transport mode. For the ASA, add an **access-list** command statement from the server to the client or vice versa. For RealPlayer, change transport mode by choosing **Options** > **Preferences** > **Transport** > **RTSP Settings**.

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

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

Restrictions and Limitations

The following restrictions apply to the RSTP inspection.

- The ASA does not support multicast RTSP or RTSP messages over UDP.
- The ASA does not have the ability to recognize HTTP cloaking where RTSP messages are hidden in the HTTP messages.
- With Cisco IP/TV, the number of translates the ASA 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.

Examples

The following example enables the RTSP inspection engine, which creates a class map to match RTSP traffic on the default ports (554 and 8554). The service policy is then applied to the outside interface.

```
ciscoasa(config)# access-list rtsp-acl permit tcp any any eq 554

ciscoasa(config)# access-list rtsp-acl permit tcp any any eq 8554
ciscoasa(config)# class-map rtsp-traffic

ciscoasa(config-cmap)# match access-list rtsp-acl

ciscoasa(config-cmap)# exit
ciscoasa(config)# policy-map rtsp_policy

ciscoasa(config-pmap)# class rtsp-traffic
ciscoasa(config-pmap-c)# inspect rtsp
ciscoasa(config-pmap-c)# exit
ciscoasa(config)# service-policy rtsp_policy interface outside
```

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
policy-map	Associates a class map with specific security actions.
service-policy	Applies a policy map to one or more interfaces.

inspect scansafe

To enable Cloud Web Security inspection on the traffic in a class, use the **inspect scansafe** command in class configuration mode. You can access the class configuration mode by first entering the **policy-map** command. To remove the inspect action, use the **no** form of this command.

inspect scansafe *scansafe_policy_name* [**fail-open** | **fail-close**]
no inspect scansafe *scansafe_policy_name* [**fail-open** | **fail-close**]

Syntax Description

scansafe_policy_name	Specifies the inspection class map name defined by the **policy-map type inspect scansafe** command.
fail-open	(Optional) Allows traffic to pass through the ASA if the Cloud Web Security servers are unavailable.
fail-close	(Optional) Drops all traffic if the Cloud Web Security servers are unavailable. **fail-close** is the default.

Command Default

fail-close is the default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Global configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
9.0(1)	This command was added.

Usage Guidelines

Cisco Cloud Web Security provides web security and web filtering services through the Software-as-a-Service (SaaS) model. Enterprises with the ASA in their network can use Cloud Web Security services without having to install additional hardware.

Note This feature is also called “ScanSafe,” so the ScanSafe name appears in some commands.

Configure this command using Modular Policy Framework:

1. Create inspection policy maps using the **policy-map type inspect scansafe** command, at least one for HTTP and one for HTTPS (assuming you want to inspect both types of traffic).
2. (Optional) Configure a whitelist using the **class-map type inspect scansafe** command.

3. Define the traffic that you want to inspect using the **class-map** command. You must configure separate class maps for HTTP and HTTPS traffic.
4. Enter the **policy-map** command to define the policy.
5. For HTTP, enter the **class** command to reference the HTTP class map.
6. Enter the **inspect scansafe** command, referencing the HTTP inspection policy map.
7. For HTTPS, enter the **class** command to reference the HTTPS class map.
8. Enter the **inspect scansafe** command, referencing the HTTPS inspection policy map.
9. Finally, apply the policy map to an interface using the **service-policy** command.

Examples

The following example configures two classes: one for HTTP and one for HTTPS. Each ACL exempts traffic to www.cisco.com and to tools.cisco.com, and to the DMZ network, for both HTTP and HTTPS. All other traffic is sent to Cloud Web Security, except for traffic from several whitelisted users and groups. The policy is then applied to the inside interface.

```
ciscoasa(config)# class-map type inspect scansafe match-any whitelist1
ciscoasa(config-cmap)# match user user1 group cisco
ciscoasa(config-cmap)# match user user2
ciscoasa(config-cmap)# match group group1
ciscoasa(config-cmap)# match user user3 group group3
ciscoasa(config)# policy-map type inspect scansafe cws_inspect_pmap1
ciscoasa(config-pmap)# parameters

ciscoasa(config-pmap-p)# http
ciscoasa(config-pmap-p)# default group default_group
ciscoasa(config-pmap-p)# class whitelist1
ciscoasa(config-pmap-c)# whitelist
ciscoasa(config)# policy-map type inspect scansafe cws_inspect_pmap2
ciscoasa(config-pmap)# parameters

ciscoasa(config-pmap-p)# https
ciscoasa(config-pmap-p)# default group2 default_group2
ciscoasa(config-pmap-p)# class whitelist1
ciscoasa(config-pmap-c)# whitelist
ciscoasa(config)# object network cisco1
ciscoasa(config-object-network)# fqdn www.cisco.com
ciscoasa(config)# object network cisco2
ciscoasa(config-object-network)# fqdn tools.cisco.com
ciscoasa(config)# object network dmz_network
ciscoasa(config-object-network)# subnet 10.1.1.0 255.255.255.0
ciscoasa(config)# access-list SCANSAFE_HTTP extended deny tcp any4 object cisco1 eq 80
ciscoasa(config)# access-list SCANSAFE_HTTP extended deny tcp any4 object cisco2 eq 80
ciscoasa(config)# access-list SCANSAFE_HTTP extended deny tcp any4 object dmz_network eq
80
ciscoasa(config)# access-list SCANSAFE_HTTP extended permit tcp any4 any4 eq 80
ciscoasa(config)# access-list SCANSAFE_HTTPS extended deny tcp any4 object cisco1 eq 443
ciscoasa(config)# access-list SCANSAFE_HTTPS extended deny tcp any4 object cisco2 eq 443
ciscoasa(config)# access-list SCANSAFE_HTTP extended deny tcp any4 object dmz_network eq
443
ciscoasa(config)# access-list SCANSAFE_HTTPS extended permit tcp any4 any4 eq 443
ciscoasa(config)# class-map cws_class1
ciscoasa(config-cmap)# match access-list SCANSAFE_HTTP
ciscoasa(config)# class-map cws_class2
ciscoasa(config-cmap)# match access-list SCANSAFE_HTTPS
```

```
ciscoasa(config)# policy-map cws_policy
ciscoasa(config-pmap)# class cws_class1
ciscoasa(config-pmap-c)# inspect scansafe cws_inspect_pmap1 fail-open
ciscoasa(config-pmap)# class cws_class2
ciscoasa(config-pmap-c)# inspect scansafe cws_inspect_pmap2 fail-open
ciscoasa(config)# service-policy cws_policy inside
```

Related Commands

Command	Description
class-map type inspect scansafe	Creates an inspection class map for whitelisted users and groups.
default user group	Specifies the default username and/or group if the ASA cannot determine the identity of the user coming into the ASA.
http[**s**] (parameters)	Specifies the service type for the inspection policy map, either HTTP or HTTPS.
inspect scansafe	Enables Cloud Web Security inspection on the traffic in a class.
license	Configures the authentication key that the ASA sends to the Cloud Web Security proxy servers to indicate from which organization the request comes.
match user group	Matches a user or group for a whitelist.
policy-map type inspect scansafe	Creates an inspection policy map so you can configure essential parameters for the rule and also optionally identify the whitelist.
retry-count	Enters the retry counter value, which is the amount of time that the ASA waits before polling the Cloud Web Security proxy server to check its availability.
scansafe	In multiple context mode, allows Cloud Web Security per context.
scansafe general-options	Configures general Cloud Web Security server options.
server {**primary** | **backup**}	Configures the fully qualified domain name or IP address of the primary or backup Cloud Web Security proxy servers.
show conn scansafe	Shows all Cloud Web Security connections, as noted by the capitol Z flag.
show scansafe server	Shows the status of the server, whether it's the current active server, the backup server, or unreachable.
show scansafe statistics	Shows total and current http connections.
user-identity monitor	Downloads the specified user or group information from the AD agent.
whitelist	Performs the whitelist action on the class of traffic.

inspect sctp

To enable or disable Stream Control Transmission Protocol (SCTP) inspection, use the **inspect sctp** command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. Use the **no** form of this command to disable SCTP inspection.

inspect sctp [*map_name*]
no inspect sctp [*map_name*]

Note SCTP inspection requires the Carrier license.

Syntax Description

map_name (Optional) Name for the SCTP inspection policy map.

Command Default

This command is disabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
9.5(2)	This command was added.

Usage Guidelines

SCTP (Stream Control Transmission Protocol) supports the telephony signaling protocol SS7 and is also a transport protocol for several interfaces in the 4G LTE mobile network architecture. You would use SCTP inspection along with GTP and Diameter inspection if you have mobile network traffic going through the device.

You can optionally specify an SCTP policy map if you want to filter on SCTP applications to provide variable services. You can selectively drop, log, or rate limit SCTP traffic classes based on the payload protocol identifier (PPID). Use the **policy-map type inspect sctp** command to create the policy map.

Examples

The following example creates an inspection policy map that will drop unassigned PPIDs (unassigned at the time this example was written), rate limit PPIDs 32-40, and log the Diameter PPID. The service policy applies the inspection to the inspection_default class, which matches all SCTP traffic.

```
policy-map type inspect sctp sctp-pmap
 match ppid 58 4294967295
```

```
  drop
 match ppid 26
  drop
 match ppid 49
  drop
 match ppid 32 40
  rate-limit 1000
 match ppid diameter
  log
policy-map global_policy
 class inspection_default
  inspect sctp sctp-pmap
!
service-policy global_policy global
```

Related Commands

Commands	Description
class	Defines the traffic class to which to apply security actions.
clear service-policy inspect sctp	Clears global SCTP statistics.
policy-map type inspect	Creates an inspection policy map.
service-policy	Applies a policy map to one or more interfaces.
show service-policy inspect sctp	Shows that status and statistics of the inspect sctp policy.

inspect sip

To enable SIP application inspection or to change the ports to which the ASA listens, use the inspect **sip** command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect sip [*sip_map*] [**tls-proxy** *proxy_name*] [**phone-proxy** *proxy_name*] [**uc-ime** *proxy_name*]
no inspect sip [*sip_map*] [**tls-proxy** *proxy_name*] [**phone-proxy** *proxy_name*] [**uc-ime** *proxy_name*]

Syntax Description

phone-proxy *proxy_name*	Enables the phone proxy for the specified inspection session.
sip_map	Specifies a SIP policy map name.
tls-proxy *proxy_name*	Enables TLS proxy for the specified inspection session. The keyword **tls-proxy** cannot be used as a Layer 7 policy map name.
uc-ime *proxy_name*	Enable the Cisco Intercompany Media Engine Proxy for SIP inspection.

Command Default

SIP inspection is enabled by default using the default inspection map, which includes the following:

- SIP instant messaging (IM) extensions: Enabled.
- Non-SIP traffic on SIP port: Permitted.
- Hide server's and endpoint's IP addresses: Disabled.
- Mask software version and non-SIP URIs: Disabled.
- Ensure that the number of hops to destination is greater than 0: Enabled.
- RTP conformance: Not enforced.
- SIP conformance: Do not perform state checking and header validation.

Also note that inspection of encrypted traffic is not enabled. You must configure a TLS proxy to inspect encrypted traffic.

The default port assignment for SIP is 5060.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added. It replaced the **fixup** command, which has been deprecated.
8.0(2)	The tls-proxy keyword was added.
9.4(1)	The **phone-proxy** and **uc-ime** keywords were removed.

Usage Guidelines

SIP is a widely used protocol for Internet conferencing, telephony, presence, events notification, and instant messaging. Partially because of its text-based nature and partially because of its flexibility, SIP networks are subject to a large number of security threats.

SIP application inspection provides address translation in message header and body, dynamic opening of ports and basic sanity checks. It also supports application security and protocol conformance, which enforce the sanity of the SIP messages, as well as detect SIP-based attacks.

SIP inspection is enabled by default. You need to configure it only if you want non-default processing, or if you want to identify a TLS proxy to enable encrypted traffic inspection.

To support SIP calls through the ASA, 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.

Limitations for SIP Inspection

SIP inspection applies NAT for embedded IP addresses. However, if you configure NAT to translate both source and destination addresses, the external address ("from" in the SIP header for the "trying" response message) is not rewritten. Thus, you should use object NAT when working with SIP traffic so that you avoid translating the destination address.

The following limitations and restrictions apply when using PAT with SIP:

- If a remote endpoint tries to register with a SIP proxy on a network protected by the ASA, the registration fails under very specific conditions, as follows:
  - PAT is configured for the remote endpoint.
  - The SIP registrar server is on the outside network.
  - The port is missing in the contact field in the REGISTER message sent by the endpoint to the proxy server.
- If a SIP device transmits a packet in which the SDP portion has an IP address in the owner/creator field (o=) 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.
- When using PAT, any SIP header field which contains an internal IP address without a port might not be translated and hence the internal IP address will be leaked outside. If you want to avoid this leakage, configure NAT instead of PAT.

Inspecting Signaling Messages

For inspecting signaling messages, the **inspect sip** command often needs to determine locations of the media endpoints (for example, IP phones).

This information is used to prepare access-control and NAT state for media traffic to traverse the firewall transparently without manual configuration.

In determining these locations, the **inspect sip** command does **not** use the tunnel default gateway route. A tunnel default gateway route is a route of the form **route** *interface* **0 0** *metric* **tunneled**. This route overrides the default route for packets that egress from IPsec tunnels. Therefore, if the **inspect sip** command is desired for VPN traffic, do not configure the tunnel default gateway route. Instead, use other static routing or dynamic routing.

Examples

The following example enables the SIP inspection engine, which creates a class map to match SIP traffic on the default port (5060). The service policy is then applied to the outside interface. To enable SIP inspection for all interfaces, use the **global** parameter in place of **interface outside**.

```
ciscoasa(config)# class-map sip-port
ciscoasa(config-cmap)# match port tcp eq 5060
ciscoasa(config-cmap)# exit
ciscoasa(config)# policy-map sip_policy

ciscoasa(config-pmap)# class sip-port
ciscoasa(config-pmap-c)# inspect sip
ciscoasa(config-pmap-c)# exit
ciscoasa(config)# service-policy sip_policy interface outside
```

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
policy-map type inspect sip	Creates an inspection policy map for SIP.
show sip	Displays information about SIP sessions established through the ASA.
show conn	Displays the connection state for different connection types.
timeout	Sets the maximum idle time duration for different protocols and session types.
tls-proxy	Defines a TLS proxy instance and sets the maximum sessions.

inspect skinny

To enable SCCP (Skinny) application inspection, use the inspect **skinny** command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect skinny [*skinny_map*] [**tls-proxy** *proxy_name*] [**phone-proxy** *proxy_name*]
no inspect skinny [*skinny_map*] [**tls-proxy** *proxy_name*] [**phone-proxy** *proxy_name*]

Syntax Description

phone-proxy *proxy_name*	Enables the phone proxy for the inspection session.
skinny_map	Specifies a skinny policy map name.
tls-proxy *proxy_name*	Enables TLS proxy for the inspection session.

Command Default

SCCP inspection is enabled by default using these defaults:

- Registration: Not enforced.
- Maximum message ID: 0x181.
- Minimum prefix length: 4
- Media timeout: 00:05:00
- Signaling timeout: 01:00:00.
- RTP conformance: Not enforced.

Also note that inspection of encrypted traffic is not enabled. You must configure a TLS proxy to inspect encrypted traffic.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added. It replaced the **fixup** command, which has been deprecated.
8.0(2)	The keyword **tls-proxy** was added.
9.4(1)	The **phone-proxy** keyword was deprecated.

Release	Modification
9.13(1)	The **tls-proxy** keyword was deprecated. The keyword will be removed in a future release.
9.14(1)	The **tls-proxy** keyword, and support for SCCP/Skinny encrypted inspection, was removed.

Usage Guidelines

Skinny (SCCP) is a simplified protocol used in VoIP networks. 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.

The ASA supports PAT and NAT for SCCP. PAT is necessary if you have more IP phones than global IP addresses for the IP phones to use. By supporting NAT and PAT of SCCP Signaling packets, Skinny application inspection ensures that all SCCP signaling and media packets can traverse the ASA.

Normal traffic between Cisco CallManager and Cisco IP Phones uses SCCP and is handled by SCCP inspection without any special configuration. The ASA also supports DHCP options 150 and 66, which it accomplishes by sending the location of a TFTP server to Cisco IP Phones and other DHCP clients. Cisco IP Phones might also include DHCP option 3 in their requests, which sets the default route.

Note The ASA supports inspection of traffic from Cisco IP Phones running SCCP protocol version 22 and earlier.

Supporting Cisco IP Phones

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. An static identity entry allows the Cisco CallManager on the higher security interface to accept registrations from the Cisco IP Phones.

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 ACL 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.

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

Restrictions and Limitations

If the address of an internal Cisco CallManager is configured for NAT or PAT to a different IP address or port, registrations for external Cisco IP Phones fail because the ASA currently does not support NAT or PAT for the file content transferred over TFTP. Although the ASA supports NAT of TFTP messages and opens a pinhole for the TFTP file, the ASA cannot translate the Cisco CallManager IP address and port embedded in the Cisco IP Phone configuration files that are transferred by TFTP during phone registration.

Note The ASA supports stateful failover of SCCP calls except for calls that are in the middle of call setup.

Inspecting Signaling Messages

For inspecting signaling messages, the **inspect skinny** command often needs to determine locations of the media endpoints (for example, IP phones).

This information is used to prepare access-control and NAT state for media traffic to traverse the firewall transparently without manual configuration.

In determining these locations, the **inspect skinny** command does **not** use the tunnel default gateway route. A tunnel default gateway route is a route of the form **route** *interface* **0 0** *metric* **tunneled**. This route overrides the default route for packets that egress from IPsec tunnels. Therefore, if the **inspect skinny** command is desired for VPN traffic, do not configure the tunnel default gateway route. Instead, use other static routing or dynamic routing.

Examples

The following example enables the SCCP inspection engine, which creates a class map to match SCCP traffic on the default port (2000). The service policy is then applied to the outside interface. To enable SCCP inspection for all interfaces, use the **global** parameter in place of **interface outside**.

```
ciscoasa(config)# class-map skinny-port
ciscoasa(config-cmap)# match port tcp eq 2000
ciscoasa(config-cmap)# exit
ciscoasa(config)# policy-map skinny_policy

ciscoasa(config-pmap)# class skinny-port
ciscoasa(config-pmap-c)# inspect skinny
ciscoasa(config-pmap-c)# exit
ciscoasa(config)# service-policy skinny_policy interface outside
```

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
policy-map type inspect skinny	Creates an inspection policy map for SCCP.
show skinny	Displays information about SCCP sessions established through the ASA.
show conn	Displays the connection state for different connection types.
timeout	Sets the maximum idle time duration for different protocols and session types.
tls-proxy	Defines a TLS proxy instance and sets the maximum sessions.

inspect snmp

To enable SNMP application inspection or to change the ports to which the ASA listens, use the inspect **snmp** command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect snmp [*map_name*]
no inspect snmp [*map_name*]

Syntax Description

map_name	The name of the SNMP map.

Command Default

This command is enabled by default starting in 9.14(1). It is disabled by default in previous releases.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added.
9.14(1)	The command enabled by default, and the SNMP map was made optional.

Usage Guidelines

Starting with 9.14(1), SNMP application inspection is applied to both to-the-device and through-the-device traffic. This inspection is necessary if you configure SNMP v3 where users are limited to specific SNMP hosts. Without the inspection, a defined v3 user can poll the device from any allowed host. SNMP inspection is enabled by default for the default ports, so you need to configure it only if you use non-default ports. The default ports are UDP/161, 162 (for all device types) and UDP/4161 for devices that also run Secure Firewall eXtensible Operating System (FXOS), as FXOS listens on UDP/161.

In releases previous to 9.14(1), SNMP inspection is not enabled by default, and it applies to through-the-box traffic only.

SNMP application inspection also lets you restrict SNMP traffic to a specific version of SNMP. Earlier versions of SNMP are less secure; therefore, denying certain SNMP versions may be required by your security policy. The system can deny SNMP versions 1, 2, 2c, or 3. To deny a specific version of SNMP, use the **deny version** command within an SNMP map, which you create using the **snmp-map** command. After configuring the SNMP map, you enable the map using the **inspect snmp** command and then apply it to one or more interfaces using the **service-policy** command.

Starting with 9.14(1), if you do not need to control the versions, simply enable SNMP inspection without a map. In previous versions, a map is required.

Examples

The following example identifies SNMP traffic, defines an SNMP map, defines a policy, enables SNMP inspection, and applies the policy to the outside interface:

```
ciscoasa(config)# access-list snmp-acl permit tcp any any eq 161

ciscoasa(config)# access-list snmp-acl permit tcp any any eq 162
ciscoasa(config)# class-map snmp-port

ciscoasa(config-cmap)# match access-list snmp-acl
ciscoasa(config-cmap)# exit
ciscoasa(config)# snmp-map inbound_snmp
ciscoasa(config-snmp-map)# deny version 1
ciscoasa(config-snmp-map)# exit
ciscoasa(config)# policy-map inbound_policy

ciscoasa(config-pmap)# class snmp-port
ciscoasa(config-pmap-c)# inspect snmp inbound_snmp

ciscoasa(config-pmap-c)# exit
```

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
deny version	Disallows traffic using a specific version of SNMP.
snmp-map	Defines an SNMP map and enables SNMP map configuration mode.
policy-map	Associates a class map with specific security actions.
service-policy	Applies a policy map to one or more interfaces.

inspect sqlnet

To enable Oracle SQL*Net application inspection, use the inspect **sqlnet** command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect sqlnet
no inspect sqlnet

Syntax Description

This command has no arguments or keywords.

Command Default

This command is enabled by default.

The default port assignment is 1521.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added. It replaced the **fixup** command, which has been deprecated.

Usage Guidelines

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

The default port assignment for SQL*Net is 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 **class-map** command to apply SQL*Net inspection to a range of port numbers.

Note Disable SQL*Net inspection when SQL data transfer occurs on the same port as the SQL control TCP port 1521. The ASA acts as a proxy when SQL*Net inspection is enabled and reduces the client window size from 65000 to about 16000 causing data transfer issues.

The ASA 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 ASA, 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 inspection engine 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.

Examples

The following example enables the SQL*Net inspection engine, which creates a class map to match SQL*Net traffic on the default port (1521). The service policy is then applied to the outside interface. To enable SQL*Net inspection for all interfaces, use the **global** parameter in place of **interface outside**.

```
ciscoasa(config)# class-map sqlnet-port
ciscoasa(config-cmap)# match port tcp eq 1521
ciscoasa(config-cmap)# exit
ciscoasa(config)# policy-map sqlnet_policy

ciscoasa(config-pmap)# class sqlnet-port
ciscoasa(config-pmap-c)# inspect sqlnet
ciscoasa(config-pmap-c)# exit
ciscoasa(config)# service-policy sqlnet_policy interface outside
```

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
policy-map	Associates a class map with specific security actions.
service-policy	Applies a policy map to one or more interfaces.
show conn	Displays the connection state for different connection types, including SQL*net.

inspect stun

To enable Session Traversal Utilities for NAT (STUN) application inspection, use the inspect **stun** command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect stun
no inspect stun

Syntax Description

This command has no arguments or keywords.

Command Default

This command is disabled by default.

The default port assignment is TCP/3478 and UDP/3478.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
9.6(2)	This command was added.

Usage Guidelines

Session Traversal Utilities for NAT (STUN), defined in RFC 5389, is used by WebRTC clients for browser-based real-time communications so that plug-ins are not necessary. WebRTC clients often use cloud STUN servers to learn their public IP addresses and ports. WebRTC uses Interactive Connectivity Establishment (ICE, RFC 5245) to verify connectivity between clients. These clients typically use UDP, although they can also use TCP or other protocols.

Because firewalls often block outgoing UDP traffic, WebRTC products such as Cisco Spark can have problems completing connections. STUN inspection opens pinholes for STUN endpoints, and enforces basic STUN and ICE compliance, to allow communications for clients if the connectivity check is acknowledged by both sides. Thus, you can avoid opening new ports in your access rules to enable these applications.

When you enable STUN inspection on the default inspection class, TCP/UDP port 3478 is watched for STUN traffic. The inspection supports IPv4 addresses and TCP/UDP only.

There are some NAT limitations for STUN inspection. For WebRTC traffic, static NAT/PAT44 are supported. Cisco Spark can support additional types of NAT, because Spark does not require pinholes. You can also use NAT/PAT64, including dynamic NAT/PAT with Cisco Spark.

STUN inspection is supported in failover and cluster modes, as pinholes are replicated. However, the transaction ID is not replicated among units. In the case where a unit fails after receiving a STUN Request and another unit received the STUN Response, the STUN Response will be dropped.

Examples

The following example enables STUN inspection as part of the default global inspection rule.

```
ciscoasa(config)# policy-map global_policy
ciscoasa(config-pmap)# class inspection_default
ciscoasa(config-pmap-c)# inspect stun
ciscoasa(config)# service-policy global_policy global
```

Related Commands

Commands	Description
class	Defines the traffic class to which to apply security actions.
policy-map	Associates a class map with specific security actions.
service-policy	Applies a policy map to one or more interfaces.
show conn	Displays the connection state for different connection types, including STUN.
show service-policy inspect diameter	Shows the status and statistics of the inspect diameter policy.

inspect sunrpc

To enable Sun RPC application inspection or to change the ports to which the ASA listens, use the inspect sunrpc command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect sunrpc
no inspect sunrpc

Syntax Description

This command has no arguments or keywords.

Command Default

This command is enabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added. It replaced the **fixup** command, which has been deprecated.

Usage Guidelines

To enable Sun RPC application inspection or to change the ports to which the ASA listens, use the inspect sunrpc command in policy map class configuration mode, which is accessible by using the **class** command within policy map configuration mode. To remove the configuration, use the **no** form of this command.

The **inspect sunrpc** command enables or disables application inspection for the Sun RPC protocol. Sun RPC is used by NFS and NIS. Sun RPC services can run on any port on the system. When a client attempts to access an Sun 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 Sun RPC program number of the service, and gets back the port number. From this point on, the client program sends its Sun RPC queries to that new port. When a server sends out a reply, the ASA intercepts this packet and opens both embryonic TCP and UDP connections on that port.

Note NAT or PAT of Sun RPC payload information is not supported.

Examples

The following example enables the RPC inspection engine, which creates a class map to match RPC traffic on the default port (111). The service policy is then applied to the outside interface. To enable RPC inspection for all interfaces, use the **global** parameter in place of **interface outside**.

```
ciscoasa(config)# class-map sunrpc-port
ciscoasa(config-cmap)# match port tcp eq 111
ciscoasa(config-cmap)# exit
ciscoasa(config)# policy-map sample_policy
ciscoasa(config-pmap)# class sunrpc-port
ciscoasa(config-pmap-c)# inspect sunrpc
ciscoasa(config-pmap-c)# exit
ciscoasa(config)# service-policy sample_policy interface outside
```

Related Commands

Commands	Description
clear configure sunrpc_server	Removes the configuration performed using the **sunrpc-server** command.
clear sunrpc-server active	Clears the pinholes that are opened by Sun RPC application inspection for specific services, such as NFS or NIS.
show running-config sunrpc-server	Displays the information about the Sun RPC service table configuration.
sunrpc-server	Allows pinholes to be created with a specified timeout for Sun RPC services, such as NFS or NIS.
show sunrpc-server active	Displays the pinholes open for Sun RPC services.

inspect tftp

To disable TFTP application inspection, or to enable it if it has been previously disabled, use the inspect **tftp** command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect tftp
no inspect tftp

Syntax Description

This command has no arguments or keywords.

Command Default

This command is enabled by default.

The default port assignment is 69.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added. It replaced the **fixup** command, which has been deprecated.

Usage Guidelines

Trivial File Transfer Protocol (TFTP), described in RFC 1350, is a simple protocol to read and write files between a TFTP server and client.

The ASA inspects TFTP traffic and dynamically creates connections and translations, if necessary, to permit file transfer between a TFTP client and server. Specifically, the inspection engine inspects TFTP read request (RRQ), write request (WRQ), and error notification (ERROR).

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

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

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

Examples

The following example enables the TFTP inspection engine, which creates a class map to match TFTP traffic on the default port (69). The service policy is then applied to the outside interface. To enable TFTP inspection for all interfaces, use the **global** parameter in place of **interface outside**.

```
ciscoasa(config)# class-map tftp-port
ciscoasa(config-cmap)# match port udp eq 69
ciscoasa(config-cmap)# exit
ciscoasa(config)# policy-map tftp_policy

ciscoasa(config-pmap)# class tftp-port
ciscoasa(config-pmap-c)# inspect tftp
ciscoasa(config-pmap-c)# exit
ciscoasa(config)# service-policy tftp_policy interface outside
```

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
policy-map	Associates a class map with specific security actions.
service-policy	Applies a policy map to one or more interfaces.

inspect vxlan

To enable Virtual Extensible Local Area Network (VXLAN) application inspection, use the **inspect vxlan** command in class configuration mode. The class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect vxlan
no inspect vxlan

Syntax Description

This command has no arguments or keywords.

Command Default

This command is disabled by default.

The default port assignment is UDP/4789.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
9.4(1)	This command was added.

Usage Guidelines

Virtual Extensible Local Area Network (VXLAN) inspection works on VXLAN encapsulated traffic that passes through the ASA. It ensures that the VXLAN header format conforms to standards, dropping any malformed packets. VXLAN inspection is not done on traffic for which the ASA acts as a VXLAN Tunnel End Point (VTEP) or a VXLAN gateway, as those checks are done as a normal part of decapsulating VXLAN packets.

VXLAN packets are UDP, normally on port 4789. This port is part of the default-inspection-traffic class, so you can simply add VXLAN inspection to the inspection_default global service policy rule. Alternatively, you can create a class for it using port or ACL matching.

Examples

The following example enables VXLAN inspection as part of the global inspection default rule.

```
ciscoasa(config)# policy-map global_policy
ciscoasa(config-pmap)# class inspection_default
ciscoasa(config-pmap-c)# inspect vxlan
```

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
policy-map	Associates a class map with specific security actions.
service-policy	Applies a policy map to one or more interfaces.

inspect waas

To enable WAAS application inspection, use the **inspect waas** command in class configuration mode. The class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect waas
no inspect waas

Syntax Description

This command has no arguments or keywords.

Command Default

No default behaviors or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.2(1)	This command was added.

Examples

The following example shows how to enable WAAS application inspection on the default inspection class.

```
policy-map global_policy
 class inspection_default
  inspect waas
```

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
policy-map	Associates a class map with specific security actions.
service-policy	Applies a policy map to one or more interfaces.

inspect xdmcp

To enable XDMCP application inspection or to change the ports to which the ASA listens, use the **inspect xdmcp** command in class configuration mode. Class configuration mode is accessible from policy map configuration mode. To remove the configuration, use the **no** form of this command.

inspect xdmcp
no inspect xdmcp

Syntax Description

This command has no arguments or keywords.

Command Default

Starting with 9.16, this command is disabled by default. In prior releases, it was enabled by default.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Class configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
7.0(1)	This command was added. It replaced the **fixup** command, which has been deprecated.

Usage Guidelines

The inspect **xdmcp** command enables or disables application inspection for the XDMCP protocol.

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 ASA must allow the TCP back connection from the Xhosted computer. To permit the back connection, use the **established** command on the ASA. 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 XWindows session, the manager talks to the display 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 ASA can NAT if needed. XDCMP inspection does not support PAT.

Examples

The following example enables the XDMCP inspection engine, which creates a class map to match XDMCP traffic on the default port (177). The service policy is then applied to the outside interface.

To enable XDMCP inspection for all interfaces, use the **global** parameter in place of **interface outside**.

```
ciscoasa(config)# class-map xdmcp-port
ciscoasa(config-cmap)# match port tcp eq 177
ciscoasa(config-cmap)# exit
ciscoasa(config)# policy-map xdmcp_policy

ciscoasa(config-pmap)# class xdmcp-port
ciscoasa(config-pmap-c)# inspect xdmcp
ciscoasa(config-pmap-c)# exit
ciscoasa(config)# service-policy xdmcp_policy interface outside
```

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
policy-map	Associates a class map with specific security actions.
service-policy	Applies a policy map to one or more interfaces.