# Cisco IOS Firewall-SIP Enhancements ALG and AIC

Enhanced Session Initiation Protocol (SIP) inspection in the Cisco IOS firewall provides basic SIP inspect functionality (SIP packet inspection and pinholes opening) as well as protocol conformance and application security. These enhancements give you more control than in previous releases on what policies and security checks to apply to SIP traffic and the capability to filter out unwanted messages or users.

The development of additional SIP functionality in Cisco IOS software provides increased support for Cisco Call Manager (CCM), Cisco Call Manager Express (CCME), and Cisco IP-IP Gateway based voice/video systems. Application Layer Gateway (ALG), and Application Inspection and Control (AIC) SIP enhancements also support RFC 3261 and its extensions.


- Finding Feature Information, page 1
- Prerequisites for Cisco IOS Firewall-SIP Enhancements ALG and AIC, page 2
- Restrictions for Cisco IOS Firewall-SIP Enhancements ALG and AIC, page 2
- Information About Cisco IOS Firewall-SIP Enhancements ALG and AIC, page 3
- How to Configure Cisco IOS Firewall-SIP Enhancements ALG and AIC, page 4
- Configuration Examples for Cisco IOS Firewall-SIP Enhancements ALG and AIC, page 20
- Additional References, page 20
- Feature Information for Cisco IOS Firewall-SIP Enhancements ALG and AIC, page 21


## Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

# Prerequisites for Cisco IOS Firewall-SIP Enhancements ALG and AIC

The following prerequisites apply to the configuration of Cisco IOS Firewall--SIP Enhancements: ALG and AIC.

### Hardware Requirements

- One of the following router platforms:
  - Cisco 861, Cisco 881, or Cisco 881G routers
  - Cisco 1700 routers
  - Cisco 1800 routers
  - Cisco 2600 routers
  - Cisco 2800 routers
  - Cisco 3700 routers
  - Cisco 3800 routers
  - Cisco 7200 routers
  - Cisco 7300 routers

### Software Requirements

- Cisco IOS Release 12.4(15)XZ or a later release.

# Restrictions for Cisco IOS Firewall-SIP Enhancements ALG and AIC

### DNS Name Resolution

Although SIP methods can have Domain Name System (DNS) names instead of raw IP addresses, this feature currently does not support DNS names.

### Earlier Releases of Cisco IOS Software

Some Cisco IOS releases earlier than Release 12.4(15)XZ may accept the configuration commands for SIP that are shown in this document; however, those earlier versions will not function properly.

# Information About Cisco IOS Firewall-SIP Enhancements ALG and AIC

## Firewall and SIP Overviews

This section provides an overview of the Cisco IOS firewall and SIP.

### Cisco IOS Firewall

The Cisco IOS firewall extends the concept of static access control lists (ACLs) by introducing dynamic ACL entries that open on the basis of the necessary application ports on a specific application and close these ports at the end of the application session. The Cisco IOS firewall achieves this functionality by inspecting the application data, checking for conformance of the application protocol, extracting the relevant port information to create the dynamic ACL entries, and closing these ports at the end of the session. The Cisco IOS firewall is designed to easily allow a new application inspection whenever support is needed.

### Session Initiation Protocol

SIP is an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. These sessions could include Internet telephone calls, multimedia distribution, and multimedia conferences. SIP is based on an HTTP-like request/response transaction model. Each transaction consists of a request that invokes a particular method or function on the server and at least one response.

SIP invitations used to create sessions carry session descriptions that allow participants to agree on a set of compatible media types. SIP makes use of elements called proxy servers to help route requests to the user's current location, authenticate and authorize users for services, implement provider call-routing policies, and provide features to users. SIP also provides a registration function that allows users to upload their current locations for use by proxy servers. SIP runs on top of several different transport protocols.

## Firewall for SIP Functionality Description

The Firewall for SIP Support feature allows SIP signaling requests to traverse directly between gateways or through a series of proxies to the destination gateway or phone. After the initial request, if the Record-Route header field is not used, subsequent requests can traverse directly to the destination gateway address as specified in the Contact header field. Thus, the Cisco IOS firewall is aware of all surrounding proxies and gateways and allows the following functionality:

- SIP signaling responses can travel the same path as SIP signaling requests.
- Subsequent signaling requests can travel directly to the endpoint (destination gateway).
- Media endpoints can exchange data between each other.

### SIP UDP and TCP Support

RFC 3261 is the current RFC for SIP, which replaces RFC 2543. This feature supports the SIP User Datagram Protocol (UDP) and the TCP format for signaling.

# SIP Inspection

This section describes the deployment scenarios supported by the Cisco IOS Firewall--SIP, ALG, and AIC Enhancements feature.

### Cisco IOS Firewall Between SIP Phones and CCM

The Cisco IOS firewall is located between CCM or CCME and SIP phones. SIP phones are registered to CCM or CCME through the firewall, and any SIP calls from or to the SIP phones pass through the firewall.

### Cisco IOS Firewall Between SIP Gateways

The Cisco IOS firewall is located between two SIP gateways, which can be CCM, CCME, or a SIP proxy. Phones are registered with SIP gateways directly. The firewall sees the SIP session or traffic only when there is a SIP call between phones registered to different SIP gateways. In some scenarios an IP-IP gateway can also be configured on the same device as the firewall. With this scenario all the calls between the SIP gateways are terminated in the IP-IP gateway.

### Cisco IOS Firewall with Local CCME and Remote CCME/CCCM

The Cisco IOS firewall is located between two SIP gateways, which can be CCM, CCME, or a SIP proxy. One of the gateways is configured on the same device as the firewall. All the phones registered to this gateway are locally inspected by the firewall. The firewall also inspects SIP sessions between the two gateways when there is a SIP call between them. With this scenario the firewall locally inspects SIP phones on one side and SIP gateways on the other side.

### Cisco IOS Firewall with Local CCME

The Cisco IOS firewall and CCME is configured on the same device. All the phones registered to the CCME are locally inspected by the firewall. Any SIP call between any of the phones registered will also be inspected by the Cisco IOS firewall.

# How to Configure Cisco IOS Firewall-SIP Enhancements ALG and AIC

## Configuring a Policy to Allow RFC 3261 Methods

Perform this task to configure a policy to allow basic RFC 3261 methods and block extension methods.

**Note** The Cisco IOS Firewall--SIP Enhancements: ALG and AIC feature provides essential support for the new SIP methods such as UPDATE and PRACK, as CCM 5.x and CCME 4.x also use these methods.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect** *protocol-name* **match-any** *class-map-name*
4. **match request method** *method-name*
5. **exit**
6. **class-map type inspect** *protocol-name* **match-any** *class-map-name*
7. **match request method** *method-name*
8. **exit**
9. **policy-map type inspect** *protocol-name policy-map-name*
10. **class type inspect** *protocol-name class-map-name*
11. **allow**
12. **exit**
13. **class type inspect** *protocol-name class-map-name*
14. **reset**
15. **exit**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **class-map type inspect** *protocol-name* **match-any** *class-map-name*<br><br>**Example:**<br>`Router(config)# class-map type inspect sip match-any sip-class1` | Creates an inspect type class map and enters class-map configuration mode. |
| **Step 4** | **match request method** *method-name*<br><br>**Example:**<br>`Router(config-cmap)# match request method invite` | Matches RFC 3261 methods. Methods include the following:<br>• ack, bye, cancel, info, invite, message, notify, options, prack, refer, register, subscribe, update. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | **exit**<br><br>**Example:**<br><br>`Router(config-cmap)# exit` | Exits class-map configuration mode. |
| **Step 6** | **class-map type inspect** *protocol-name* **match-any** *class-map-name*<br><br>**Example:**<br><br>`Router(config)# class-map type inspect sip match-any sip-class2` | Creates an inspect type class map and enters class-map configuration mode. |
| **Step 7** | **match request method** *method-name*<br><br>**Example:**<br><br>`Router(config-cmap)# match request method message` | Matches RFC 3261 methods, which include the following:<br>• ack, bye, cancel, info, invite, message, notify, options, prack, refer, register, subscribe, update. |
| **Step 8** | **exit**<br><br>**Example:**<br><br>`Router(config-cmap)# exit` | Exits class-map configuration mode. |
| **Step 9** | **policy-map type inspect** *protocol-name policy-map-name*<br><br>**Example:**<br><br>`Router(config)# policy-map type inspect sip sip-policy` | Creates an inspect type policy map and enters policy-map configuration mode. |
| **Step 10** | **class type inspect** *protocol-name class-map-name*<br><br>**Example:**<br><br>`Router(config-pmap)# class type inspect sip sip_class1` | Specifies the class on which the action is performed and enters policy-map class configuration mode. |
| **Step 11** | **allow**<br><br>**Example:**<br><br>`Router(config-pmap-c)# allow` | Allows SIP inspection. |
| **Step 12** | **exit**<br><br>**Example:**<br><br>`Router(config-pmap-c)# exit` | Exits policy-map class configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 13** | **class type inspect** *protocol-name class-map-name*<br><br>**Example:**<br><br>`Router(config-pmap)# class type inspect sip sip-class2` | Specifies the class on which the action is performed and enters policy-map class configuration mode. |
| **Step 14** | **reset**<br><br>**Example:**<br><br>`Router(config-pmap-c)# reset` | Resets the class map. |
| **Step 15** | **exit**<br><br>**Example:**<br><br>`Router(config-pmap-c)# exit` | Exits policy-map class configuration mode. |

# Configuring a Policy to Block Messages

Perform this task to configure a policy to block SIP messages coming from a particular proxy device.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type regex** *parameter-map-name*
4. **pattern** *url-pattern*
5. **exit**
6. **class-map type inspect** *protocol-name class-map-name*
7. **match request header** *field* **regex** *regex-param-map*
8. **exit**
9. **policy-map type inspect** *protocol-name policy-map-name*
10. **class type inspect** *protocol-name class-map-name*
11. **reset**
12. **exit**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable** | Enables privileged EXEC mode. |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br>`Router> enable` | • Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **parameter-map type regex** *parameter-map-name*<br>**Example:**<br>`Router(config)# parameter-map type regex unsecure-proxy` | Configures a parameter-map type to match a specific traffic pattern and enters profile configuration mode. |
| **Step 4** | **pattern** *url-pattern*<br>**Example:**<br>`Router(config-profile)# pattern “compromised.server.com”` | Matches a call based on the SIP uniform resource identifier (URI). |
| **Step 5** | **exit**<br>**Example:**<br>`Router(config-profile)# exit` | Exits profile configuration mode. |
| **Step 6** | **class-map type inspect** *protocol-name class-map-name*<br>**Example:**<br>`Router(config)# class-map type inspect sip sip-class` | Creates an inspect type class map and enters class-map configuration mode. |
| **Step 7** | **match request header** *field* **regex** *regex-param-map*<br>**Example:**<br>`Router(config-cmap)# match request header Via regex unsecure-proxy` | Configures a class-map type to match a specific request header pattern. |
| **Step 8** | **exit**<br>**Example:**<br>`Router(config-cmap)# exit` | Exits class-map configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 9** | **policy-map type inspect** *protocol-name policy-map-name*<br>**Example:**<br>`Router(config)# policy-map type inspect sip sip-policy` | Creates an inspect type policy map and enters policy-map configuration mode. |
| **Step 10** | **class type inspect** *protocol-name class-map-name*<br>**Example:**<br>`Router(config-pmap)# class type inspect sip sip-class` | Specifies the class on which the action is performed and enters policy-map class configuration mode. |
| **Step 11** | **reset**<br>**Example:**<br>`Router(config-pmap-c)# reset` | Resets the class map. |
| **Step 12** | **exit**<br>**Example:**<br>`Router(config-pmap-c)# exit` | Exits policy-map class configuration mode. |

# Configuring a 403 Response Alarm

Perform this task to configure a policy to generate an alarm whenever a 403 response is returned.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type regex** *parameter-map-name*
4. **pattern** *url-pattern*
5. **exit**
6. **class-map type inspect** *protocol-name class-map-name*
7. **match response status regex** *regex-parameter-map*
8. **exit**
9. **policy-map type inspect** *protocol-name policy-map-name*
10. **class type inspect** *protocol-name class-map-name*
11. **log**
12. **exit**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br><br>`Router> enable` | Enables privileged EXEC mode.<br><br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br><br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **parameter-map type regex** *parameter-map-name*<br><br>**Example:**<br><br>`Router(config)# parameter-map type regex allowed-im-users` | Configures a parameter-map type to match a specific traffic pattern and enters profile configuration mode. |
| **Step 4** | **pattern** *url-pattern*<br><br>**Example:**<br><br>`Router(config-profile)# pattern “403”` | Matches a call based on the SIP URI. |
| **Step 5** | **exit**<br><br>**Example:**<br><br>`Router(config-profile)# exit` | Exits profile configuration mode. |
| **Step 6** | **class-map type inspect** *protocol-name class-map-name*<br><br>**Example:**<br><br>`Router(config)# class-map type inspect sip sip-class` | Creates an inspect type class map and enters class-map configuration mode. |
| **Step 7** | **match response status regex** *regex-parameter-map*<br><br>**Example:**<br><br>`Router(config-cmap)# match response status regex allowed-im-users` | Configures a class-map type to match a specific response pattern. |
| **Step 8** | **exit**<br><br>**Example:**<br><br>`Router(config-cmap)# exit` | Exits class-map configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 9** | **policy-map type inspect** *protocol-name policy-map-name*<br><br>**Example:**<br><br>`Router(config)# policy-map type inspect sip sip-policy` | Creates an inspect type policy map and enters policy-map configuration mode. |
| **Step 10** | **class type inspect** *protocol-name class-map-name*<br><br>**Example:**<br><br>`Router(config-pmap)# class type inspect sip sip-class` | Specifies the class on which the action is performed and enters policy-map class configuration mode. |
| **Step 11** | **log**<br><br>**Example:**<br><br>`Router(config-pmap-c)# log` | Generates a log of messages. |
| **Step 12** | **exit**<br><br>**Example:**<br><br>`Router(config)# exit` | Exits policy-map class configuration mode. |

# Limiting Application Messages

Perform this task to configure a policy to rate-limit INVITE messages.

**Note** While configuring the **rate-limit** command, do not configure the **allow** or **reset** commands. An error message is displayed if you try to configure the **allow** or **reset** commands while configuring the **rate-limit** command and vice versa.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect** *protocol-name* **match-any** *class-map-name*
4. **match request method** *method-name*
5. **exit**
6. **policy-map type inspect** *protocol-name policy-map-name*
7. **class type inspect** *protocol-name class-map-name*
8. **rate-limit** *limit-number*
9. **exit**
10. **exit**
11. **class-map type inspect match-any** *class-map-name*
12. **match protocol** *protocol-name*
13. **exit**
14. **policy-map type inspect** *policy-map-name*
15. **class type inspect** *class-map-name*
16. **inspect**
17. **service-policy** *protocol-name policy-map-name*
18. **exit**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **class-map type inspect** *protocol-name* **match-any** *class-map-name*<br><br>**Example:**<br>`Router(config)# class-map type inspect sip match-any class-2` | Creates an inspect type class map and enters class-map configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | **match request method** *method-name*<br>**Example:**<br>`Router(config-cmap)# match request method invite` | Matches RFC 3261 methods. Methods include the following:<br>• ack, bye, cancel, info, invite, message, notify, options, prack, refer, register, subscribe, update. |
| **Step 5** | **exit**<br>**Example:**<br>`Router(config-cmap)# exit` | Exits class-map configuration mode. |
| **Step 6** | **policy-map type inspect** *protocol-name policy-map-name*<br>**Example:**<br>`Router(config)# policy-map type inspect sip policy-2` | Creates an inspect type policy map and enters policy-map configuration mode. |
| **Step 7** | **class type inspect** *protocol-name class-map-name*<br>**Example:**<br>`Router(config-pmap)# class type inspect sip class-2` | Specifies the class on which the action is performed and enters policy-map class configuration mode. |
| **Step 8** | **rate-limit** *limit-number*<br>**Example:**<br>`Router(config-pmap-c)# rate-limit 16` | Limits the number of SIP messages that strike the Cisco IOS firewall every second. |
| **Step 9** | **exit**<br>**Example:**<br>`Router(config-pmap-c)# exit` | Exits policy-map class configuration mode. |
| **Step 10** | **exit**<br>**Example:**<br>`Router(config-pmap)# exit` | Exits policy-map configuration mode and enters global configuration mode. |
| **Step 11** | **class-map type inspect match-any** *class-map-name*<br>**Example:**<br>`Router(config)# class-map type inspect match-any class-1` | Creates an inspect type class map and enters class-map configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 12** | **match protocol** *protocol-name*<br>**Example:**<br>`Router(config-cmap)# match protocol sip` | Configures the match criterion for a class map on the basis of the specified protocol. |
| **Step 13** | **exit**<br>**Example:**<br>`Router(config-cmap)# exit` | Exits class-map configuration mode. |
| **Step 14** | **policy-map type inspect** *policy-map-name*<br>**Example:**<br>`Router(config)# policy-map type inspect policy-1` | Creates an inspect type policy map and enters policy-map configuration mode. |
| **Step 15** | **class type inspect** *class-map-name*<br>**Example:**<br>`Router(config-pmap)# class type inspect class-1` | Specifies the class on which the action is performed and enters policy-map class configuration mode. |
| **Step 16** | **inspect**<br>**Example:**<br>`Router(config-pmap-c)# inspect` | Enables stateful packet inspection. |
| **Step 17** | **service-policy** *protocol-name policy-map-name*<br>**Example:**<br>`Router(config-pmap-c)# service-policy sip policy_2` | Attaches the policy map to the service policy for the interface or virtual circuit. |
| **Step 18** | **exit**<br>**Example:**<br>`Router(config-pmap-c)# exit` | Exits policy-map class configuration mode. |

# Limiting Application Messages for a Particular Proxy

Perform this task to configure a policy to rate-limit INVITE messages coming for a particular proxy.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type regex** *parameter-map-name*
4. **pattern** *url-pattern*
5. **exit**
6. **class-map type inspect** *protocol-name* **match-any** *class-map-name*
7. **match request method** *method-name*
8. **match request header** *field* **regex** *regex-param-map*
9. **exit**
10. **policy-map type inspect** *protocol-name policy-map-name*
11. **class type inspect** *protocol-name class-map-name*
12. **rate-limit** *limit-number*
13. **exit**
14. **exit**
15. **class-map type inspect match-any** *class-map-name*
16. **match protocol** *protocol-name*
17. **exit**
18. **policy-map type inspect** *policy-map-name*
19. **class type inspect** *class-map-name*
20. **inspect**
21. **service-policy** *protocol-name policy-map-name*
22. **exit**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **parameter-map type regex** *parameter-map-name*<br>**Example:**<br>`Router(config)# parameter-map type regex rate-limited-proxy` | Configures a parameter-map type to match a specific traffic pattern and enters profile configuration mode. |
| **Step 4** | **pattern** *url-pattern*<br>**Example:**<br>`Router(config-profile)# pattern “compromised.server.com”` | Matches a call based on the SIP URI. |
| **Step 5** | **exit**<br>**Example:**<br>`Router(config-cmap)# exit` | Exits profile configuration mode. |
| **Step 6** | **class-map type inspect** *protocol-name* **match-any** *class-map-name*<br>**Example:**<br>`Router(config)# class-map type inspect sip match-any class_2` | Creates an inspect type class map and enters class-map configuration mode. |
| **Step 7** | **match request method** *method-name*<br>**Example:**<br>`Router(config-cmap)# match request method invite` | Matches RFC 3261 methods. Methods include the following:<br>• ack, bye, cancel, info, invite, message, notify, options, prack, refer, register, subscribe, update. |
| **Step 8** | **match request header** *field* **regex** *regex-param-map*<br>**Example:**<br>`Router(config-cmap)# match request header Via regex rate-limited-proxy` | Configures a class-map type to match a specific request header pattern. |
| **Step 9** | **exit**<br>**Example:**<br>`Router(config-cmap)# exit` | Exits class-map configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 10** | **policy-map type inspect** *protocol-name policy-map-name*<br>**Example:**<br>`Router(config)# policy-map type inspect sip policy-2` | Creates an inspect type policy map and enters policy-map configuration mode. |
| **Step 11** | **class type inspect** *protocol-name class-map-name*<br>**Example:**<br>`Router(config-pmap)# class type inspect sip class-2` | Specifies the class on which the action is performed and enters policy-map class configuration mode. |
| **Step 12** | **rate-limit** *limit-number*<br>**Example:**<br>`Router(config-pmap-c)# rate-limit 16` | Limits the number of SIP messages that strike the Cisco IOS firewall every second. |
| **Step 13** | **exit**<br>**Example:**<br>`Router(config-pmap-c)# exit` | Exits policy-map class configuration mode. |
| **Step 14** | **exit**<br>**Example:**<br>`Router(config-pmap)# exit` | Exits policy-map configuration mode and enters global configuration mode. |
| **Step 15** | **class-map type inspect match-any** *class-map-name*<br>**Example:**<br>`Router(config)# class-map type inspect match-any class-1` | Creates an inspect type class map and enters class-map configuration mode. |
| **Step 16** | **match protocol** *protocol-name*<br>**Example:**<br>`Router(config-cmap)# match protocol sip` | Configures the match criterion for a class map on the basis of the specified protocol. |
| **Step 17** | **exit**<br>**Example:**<br>`Router(config-cmap)# exit` | Exits class-map configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 18** | **policy-map type inspect** *policy-map-name*<br>**Example:**<br>`Router(config)# policy-map type inspect policy-1` | Creates an inspect type policy map and enters policy-map configuration mode. |
| **Step 19** | **class type inspect** *class-map-name*<br>**Example:**<br>`Router(config-pmap)# class type inspect class-1` | Specifies the class on which the action is performed and enters policy-map class configuration mode. |
| **Step 20** | **inspect**<br>**Example:**<br>`Router(config-pmap-c)# inspect` | Enables stateful packet inspection. |
| **Step 21** | **service-policy** *protocol-name policy-map-name*<br>**Example:**<br>`Router(config-pmap-c)# service-policy sip policy-2` | Attaches the policy map to the service policy for the interface or virtual circuit. |
| **Step 22** | **exit**<br>**Example:**<br>`Router(config-pmap-c)# exit` | Exits policy-map class configuration mode. |

# Verifying and Troubleshooting Cisco IOS Firewall-SIP Enhancements ALG and AIC

The following commands can be used to troubleshoot the Cisco IOS Firewall--SIP Enhancements: ALG and AIC feature:

1. **clear zone-pair**
2. **debug cce**
3. **debug ip inspect**
4. **debug policy-map type inspect**
5. **show policy-map type inspect zone-pair**
6. **show zone-pair security**

**Note** Effective with Cisco IOS Release 12.4(20)T, the **debug ip inspect** command is replaced by the **debug policy-firewall** command. See the *Cisco IOS Debug Command Reference* for more information.

## Examples

The following is sample output of the **show policy-map type inspect zone-pair** command when the **session** keyword is used.

```
Router# show policy-map type inspect zone-pair session
policy exists on zp zp_test_out_self
 Zone-pair: zp_test_out_self
  Service-policy inspect : test
   Class-map: c_sip (match-any)
   ...
    Number of Established Sessions = 2
    Established Sessions
     Session 6717A7A0 (192.168.105.118:62265)=>(192.168.105.2:5060) sip:udp SIS_OPEN
      Created 00:10:27, Last heard 00:00:03
      Bytes sent (initiator:responder) [35579:14964]
     Session 67179EA0 (192.168.105.119:62266)=>(192.168.105.2:5060) sip:udp SIS_OPEN
      Created 00:10:27, Last heard 00:03:17
      Bytes sent (initiator:responder) [10689:4093]
    Number of Pre-generated Sessions = 7
     Pre-generated Sessions
      Pre-gen session 6717A560 192.168.105.2[1024:65535]=>192.168.105.118[62265:62265]
   sip:udp
       Created never, Last heard never
       Bytes sent (initiator:responder) [0:0]
      Pre-gen session 67179C60 192.168.105.2[1024:65535]=>192.168.105.119[62266:62266]
   sip:udp
       Created never, Last heard never
       Bytes sent (initiator:responder) [0:0]
      Pre-gen session 67176F60 192.168.105.118[1024:65535]=>192.168.105.2[5060:5060]
 sip:udp
       Created never, Last heard never
       Bytes sent (initiator:responder) [0:0]
      Pre-gen session 67176AE0 192.168.105.118[1024:65535]=>192.168.105.2[18318:18318]
   sip-RTP-data:udp
       Created never, Last heard never
       Bytes sent (initiator:responder) [0:0]
      Pre-gen session 671768A0 192.168.105.2[1024:65535]=>192.168.105.118[62495:62495]
   sip-RTP-data:udp
       Created never, Last heard never
       Bytes sent (initiator:responder) [0:0]
      Pre-gen session 671783A0 192.168.105.118[1024:65535]=>192.168.105.2[18319:18319]
   sip-RTCP-data:udp
       Created never, Last heard never
       Bytes sent (initiator:responder) [0:0]
      Pre-gen session 67176420 192.168.105.2[1024:65535]=>192.168.105.118[62496:62496]
   sip-RTCP-data:udp
       Created never, Last heard never
       Bytes sent (initiator:responder) [0:0]
```

The following is sample output of the **show zone-pair security** command.

```
Router# show zone-pair security
Zone-pair name zp_in_out
 Source-Zone inside Destination-Zone outside
 service-policy test
Zone-pair name zp_in_self
 Source-Zone inside Destination-Zone self
 service-policy test
Zone-pair name zp_self_out
```

```
Source-Zone self Destination-Zone outside
service-policy test
```

# Configuration Examples for Cisco IOS Firewall-SIP Enhancements ALG and AIC

## Example Firewall and SIP Configuration

The following example shows how to configure the Cisco IOS Firewall--SIP Enhancements: ALG and AIC feature when the Cisco IOS firewall is located between two SIP gateways (CCM or CCME), as described in the Cisco IOS Firewall Between SIP Gateways. Some phones are registered to the CCME inside the firewall (inside zone). Other phones are registered to another CCME / CCM outside the firewall (outside zone). Cisco IOS firewall is configured for SIP inspection when there is no IP-IP gateway configured on the firewall device.

```
class-map type inspect sip match-any sip-aic-class
match request method invite
policy-map type inspect sip sip-aic-policy
class type inspect sip sip-aic-class
rate-limit 15
!
policy-map type inspect sip-policy
class type inspect sip-traffic-class
service-policy sip sip-aic-policy
!
class-map type inspect match-any sip-traffic-class
match protocol sip
!
policy-map type inspect sip-policy
class type inspect sip-traffic-class
inspect my-parameters
!
zone security inside
zone security outside
!
interface fastethernet 0
zone-member security inside
interface fastethernet 1
zone-member security outside
!
zone-pair security in-out source inside destination outside
service-policy type inspect sip-policy
!
zone-pair security in-self source inside destination self
service-policy type inspect sip-policy
```

# Additional References

### Related Documents

| Related Topic | Document Title |
|---|---|
| Cisco IOS commands | Cisco IOS Master Commands List, All Releases |
| Cisco IOS firewall commands | *Cisco IOS Security Command Reference* |

| Related Topic | Document Title |
|---|---|
| SIP information and configuration tasks | Configuring Session Initiation Protocol for Voice over IP" module in the *Cisco IOS Voice, Video, and Fax Configuration Guide* |
| Additional SIP Information | Guide to Cisco Systems VoIP Infrastructure Solution for SIP |

### MIBs

| MIB | MIBs Link |
|---|---|
| None | To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:<br>http://www.cisco.com/go/mibs |

### RFCs

| RFC | Title |
|---|---|
| RFC 3261 | *SIP: Session Initiation Protocol* |

### Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

# Feature Information for Cisco IOS Firewall-SIP Enhancements ALG and AIC

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

***Table 1: Feature Information for Cisco IOS Firewall-SIP Enhancements: ALG and AIC***

| Feature Name | Releases | Feature Information |
|---|---|---|
| Cisco IOS Firewall--SIP Enhancements: ALG and AIC | 12.4(15)XZ 12.4(20)T | This feature provides voice security enhancements within the firewall feature set in Cisco IOS software for Release 12.4(15)XZ and later releases.<br><br>In Release 12.4(15)XZ, this feature was introduced on the Cisco 861, Cisco 881, and Cisco 881G routers.<br><br>In Release 12.4(20)T, this feature was implemented on the Cisco 1700, Cisco 1800, Cisco 2600, Cisco 2800, Cisco 3700, Cisco 3800, Cisco 7200, and Cisco 7300 routers.<br><br>The following commands were introduced or modified: **class-map type inspect**, **match protocol**, **match protocol-violation**, **match req-resp**, **match request**, **match response**, **policy-map type inspect**, **rate-limit (firewall)**. |