Application Inspection and Control for SMTP

Last Updated: January 20, 2012

The Application Inspection for SMTP feature provides an intense provisioning mechanism that can be configured to inspect packets on a granular level so that malicious network activity, related to the transfer of e-mail at the application level, can be identified and controlled. This feature qualifies the Cisco IOS firewall extended Simple Mail Transfer Protocol (ESMTP) module as an "SMTP application firewall," which protects in a similar way to that of an HTTP application firewall.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 at the end of this document.

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 Application Inspection and Control for SMTP

Follow the appropriate configuration tasks outlined in the Zone-Based Policy Firewall module before configuring the Application Inspection and Control for SMTP feature. This module contains important information about class-maps and policy-maps and their associated “match” statements necessary for configuring an SMTP policy.

SMTP Policy Requirements

Both SMTP and ESMTP inspection provide a basic method for exchanging e-mail messages between the client and server to negotiate capabilities and use these capabilities in an e-mail transaction. An ESMTP session is similar to an SMTP session, except for one difference--the Extended HELO (EHLO) command. The EHLO command is sent by a client to initiate the capability dialogue. After the client receives a successful response to the EHLO command, the client works the same way as SMTP, except that the client may issue new extended commands, and it may add a few parameters to the MAIL FROM and REPT TO commands.

Previously, if the Cisco IOS software was configured to inspect SMTP session only, inspection was configured by entering the **match protocol smtp** command. This action would “mask” the EHLO command to prevent capability negotiation and cause the client to go back to the HELO command and basic SMTP.

To have a workable policy for both ESMTP and SMTP inspection, the **match protocol smtp** command must be configured in the top-level policy before the Application Inspection and Control for SMTP features are implemented. See the Configuring a Default Policy for Application Inspection task for more information.

The SMTP policy (which specifies the particular SMTP configuration) is included as a child-policy in the top-level “inspect” policy-map. See the “Top-level Class Maps and Policy Maps” section in the Zone-Based Policy Firewall module for more information.

Restrictions for Application Inspection and Control for SMTP

The Application Inspection and Control for SMTP feature has the following restrictions:

- The **match cmd-line length gt** command filter can co-exist only with a **match cmd verb** command filter in the SMTP match-all class -map (**class-map type inspect smtp**). Any attempt to pair the **match cmd-line length gt** command filter with any other filter is not allowed by the CLI.
- The alternative data transfer SMTP command extension BDAT is not supported. This command is substituted for the DATA command while the SMTP body is transferred. The BDAT command extension is used by the Cisco IOS firewall to mask the CHUNKING keyword in the EHLO response to the Application Inspection and Control for SMTP feature, preventing a client from using it.
- The “mask” action can be configured only with a class having either or both of the **match cmd verb** or **match ehlo reply** commands. This action cannot be configured with a class having any other filter.

Information About Application Inspection and Control for SMTP

The Application Inspection and Control for SMTP feature inspects SMTP in a granular way and is complemented by an intensive provisioning system to help filter e-mail.

Benefits of Application Inspection and Control for SMTP

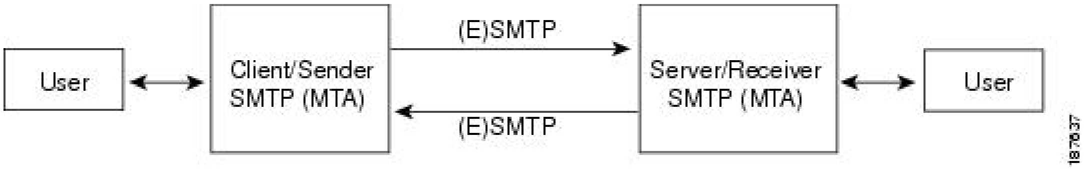
The Application Inspection and Control for SMTP feature provides the following benefits:

- E-mail senders and user accounts are restricted to filter spam e-mail from suspected domains.
- An action can be specified, which occurs when a number of invalid recipients appears on an SMTP connection. This action helps identify spammers who are looking for valid user accounts.
- The number of invalid SMTP recipients can be restricted by specifying a maximum number for invalid recipients on an SMTP connection.
- A pattern can be specified that identifies e-mail addressed to a particular recipient or domain in cases where a server is functioning as a relay.
- A provisioning mechanism that provides masks specified verbs in an SMTP connection to block potentially dangerous SMTP commands.
- The maximum length value for the SMTP e-mail header can be specified to prevent a Denial of Service (DoS) attack (also called a buffer overflow attack). A DoS attack occurs when the attacker continuously sends a large number of incomplete IP fragments, causing the firewall to lose time and memory while trying to reassemble the fake packets.
- The maximum length of an SMTP command line can be specified to prevent a DoS attack.
- Multipurpose Internet Mail Extension (MIME) content file-types (text, HTML, images, applications, documents,and so on) can be restricted in the body of the e-mail from being transmitted over SMTP.
- Unknown content-encoding types can be restricted from being transmitted over SMTP.
- Specified content-types and content encoding types can be restricted in the SMTP e-mail body.
- Monitor arbitrary patterns (text strings) in the SMTP e-mail message header (subject field) or body.
- A parameter in an EHLO server reply and mask can be specified to prevent a sender (client) from using the service extension in the server reply.
- An SMTP connection can be dropped with an SMTP sender (client) if the SMTP connection violates the specified policy.
- SMTP commands or the parameters returned by the server in response to an EHLO command can be explicitly masked by specifying these SMTP commands.
- An action can be logged for a class type in an SMTP policy-map.

Cisco Common Classification Policy Language

The Cisco Common Classification Policy Language (C3PL) CLI structure is used to provision ESMTP inspection. ESMTP is provisioned by defining a match criterion on an SMTP class-map and associate actions to the match criterion defined in the SMTP policy-map. The Application Inspection and Control for SMTP feature adds new match criteria and actions to the existing SMTP policy maps that are discussed in the Zone-Based Policy Firewall module, which describes the Cisco IOS unidirectional firewall policy between groups of interfaces known as zones.

Figure 1 ESMTP Communication Between a Sender and Receiver



Common Classification Engine SMTP Database and Action Module

The Common Classification Engine (CCE) SMTP database is the site at which manually configured policy information is processed and converted into signatures. The information in these signatures is put into regular expression tables, which are then used to parse packets as they are switched by a router.

The SMTP database has two interfaces. One interface has the control plane, which is used to accept user configured policies, and the other interface has the CCE data-plane engine, which is used to classify a packet.

An action module is used as a part of the Context-Based Access Control (CBAC) SMTP inspection module to organize and trigger SMTP inspection. CBAC is used to detect and block SMTP attacks (illegal SMTP commands) and sends notifications when SMTP attacks occur.

How to Configure Application Inspection and Control for SMTP

Configuring a Default Policy for Application Inspection

If no policy is configured for SMTP, then there is no application inspection for SMTP. The firewall creates a TCP session and only performs “pinholing,” which allows an application to have access to the protected network. Having an open gap in a firewall can expose the protected system to malicious abuse. The steps below are used to provide minimum application inspection protections for SMTP by enforcing the EHLO and HELO SMTP commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect smtp** *class-map-name*
4. **match protocol smtp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect smtp** *class-map-name* **Example:** `Router(config)# class-map type inspect smtp c1`	Creates a class map for the SMTP protocol and enters class-map configuration mode.
Step 4	**match protocol smtp** **Example:** `Router(config-cmap)# match protocol smtp`	Enables inspection for ESMTP and SMTP.

Restricting Spam from a Suspicious E-Mail Sender Address or Domain

An e-mail sender and user accounts can be restricted to filter spam e-mail from suspected domains. Spam is restricted by using the **match sender address regex** command to match the parameter-map name of a specific traffic pattern that specifies a sender domain or e-mail address in the SMTP traffic. The specified pattern is scanned in the parameter for the SMTP **MAIL FROM:** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type regex** *parameter-map-name*
4. **pattern** *traffic-pattern*
5. exit
6. **class-map type inspect smtp match-any** *class-map-name*
7. **match sender address regex** *parameter-map-name*
8. exit
9. **policy-map type inspect smtp** *policy-map-name*
10. **class type inspect smtp** *class-map-name*
11. **log**
12. **reset**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type regex** *parameter-map-name* **Example:** `Router(config)# parameter-map type regex bad-guys`	Enter the parameter-map name of a specific traffic pattern. Once the parameter-map name is specified, parameter-map profile configuration mode is entered.

	Command or Action	Purpose

Step 4	**pattern** *traffic-pattern*
	Example:
	`Router(config-profile)# pattern “*deals\.com”`
	Example:
	`Router(config-profile)# pattern “*crazyperson*@wrdmail\.com”`
		Specifies the Cisco IOS regular expression (regex) pattern that matches the traffic pattern for the e-mail sender or user accounts from suspected domains that are causing the spam e-mail.

Step 5	exit	Exits parameter-map profile configuration mode.
Step 6	**class-map type inspect smtp match-any** *class-map-name* **Example:** `Router(config)# class-map type inspect smtp match-any c1`	Creates a class map for the SMTP protocol so the match criteria is set to match any criteria for this class map and enters class-map configuration mode.
Step 7	**match sender address regex** *parameter-map-name* **Example:** `Router(config-cmap)# match sender address regex bad-guys`	Enters the parameter-map name class, which was defined in Step 3, to specify the Cisco IOS regular expression (regex) patterns for the class-map.
Step 8	exit	Exits class-map configuration mode.
Step 9	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.
Step 10	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures SMTP inspection parameters for this class map.
Step 11	**log** **Example:** `Router(config-pmap)# log`	Logs an action related to this class-type in the SMTP policy map.

	Command or Action	Purpose
Step 12	**reset** **Example:** `Router(config-pmap)# reset`	(Optional) Drops an SMTP connection with an SMTP sender (client) if it violates the specified policy. This action sends an error code to the sender and closes the connection gracefully.

Identifying and Restricting Spammers Searching for User Accounts in a Domain

Spammers who search for a large number of user accounts in a domain typically send the same e-mail to all the user accounts they find in this domain. Spammers can be identified and restricted from searching for user accounts in a domain by using the **match recipient count gt** command to specify an action that occurs when a number of invalid recipients appear on an SMTP connection.

Note The **match recipient count gt** command does not count the number of recipients specified in the To or Cc fields in the e-mail header.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect smtp** *class-map-name*
4. **match recipient count gt** *value*
5. exit
6. **policy-map type inspect smtp** *policy-map-name*
7. **class type inspect smtp** *class-map-name*
8. **reset**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect smtp** *class-map-name* **Example:** `Router(config)# class-map type inspect smtp c1`	Creates a class map for the SMTP protocol and enters class-map configuration mode.
Step 4	**match recipient count gt** *value* **Example:** `Router(config-cmap)# match recipient count gt 25`	Sets a limit on the number of RCPT SMTP commands sent by the sender (client) to recipients who are specified in a single SMTP transaction. This command determines the number of RCPT lines and invalid recipients (for which the server has replied “500 No such address”) in the SMTP transaction.
Step 5	exit	Exits class-map configuration mode.
Step 6	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode. • The *policy-map-name* argument is the name of the policy map.
Step 7	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures SMTP inspection parameters for this class map.
Step 8	**reset** **Example:** `Router(config-pmap)# reset`	(Optional) Drops an SMTP connection with an SMTP sender (client) if it violates the specified policy. This action sends an error code to the sender and closes the connection gracefully.

Restricting the Number of Invalid SMTP Recipients

If a sender specifies in an invalid e-mail recipient and SMTP encounters this invalid recipient on the SMTP connection, then SMTP sends an error code reply to the e-mail sender (client) to specify another recipient. In this case, the event did not violate the SMTP protocol or indicate that this particular SMTP connection is bad. However, if a pattern of invalid recipients appears, then a reasonable threshold can be set to restrict these nuisance SMTP connections. The **match recipient invalid count gt** command is used to help identify and restrict the number of invalid SMTP recipients that can appear in an e-mail from senders who try common names on a domain in the hope that they discover a valid username to whom they can send spam.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect smtp** *class-map-name*
4. **match recipient invalid count gt** *value*
5. exit
6. **policy-map type inspect smtp** *policy-map-name*
7. **class type inspect smtp** *class-map-name*
8. **reset**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect smtp** *class-map-name* **Example:** `Router(config)# class-map type inspect smtp c1`	Creates a class map for the SMTP protocol and enters class-map configuration mode.
Step 4	**match recipient invalid count gt** *value* **Example:** `Router(config-cmap)# match recipient invalid count gt 5`	Specifies a maximum number of invalid e-mail recipients on this SMTP connection.
Step 5	exit	Exits class-map configuration mode.

	Command or Action	Purpose
Step 6	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.
Step 7	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures SMTP inspection parameters for this class map.
Step 8	**reset** **Example:** `Router(config-pmap)# reset`	(Optional) Drops an SMTP connection with an SMTP sender (client) if it violates the specified policy. This action sends an error code to the sender and closes the connection gracefully.

Specifying a Recipient Pattern to Learn Spam Senders and Domain Information

A nonexistent e-mail recipient pattern can be specified to learn about spam senders and their domain information by luring them to use this nonexistent e-mail recipient pattern. This pattern is a regular-expression (regex) that can be specified to identify an e-mail addressed to a particular recipient or domain when a server is functioning as a relay. The specified pattern is checked in the SMTP RCPT command (SMTP envelope) parameter to identify if the recipient is either used as an argument or a source-list to forward mail in the route specified in the list.

Note The **match recipient address regex** command does not operate on the To or Cc fields in the e-mail header.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type regex** *parameter-map-name*
4. **pattern** *traffic-pattern*
5. exit
6. **class-map type inspect smtp** *class-map-name*
7. **match recipient address regex** *parameter-map-name*
8. exit
9. **policy-map type inspect smtp** *policy-map-name*
10. **class type inspect smtp** *class-map-name*
11. **log**
12. **reset**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type regex** *parameter-map-name* **Example:** `Router(config)# parameter-map type regex known-unknown-users`	Enter the parameter-map name of a specific traffic pattern. Once the parameter-map name is specified, parameter-map profile configuration mode is entered.
Step 4	**pattern** *traffic-pattern* **Example:** `Router(config-profile)# pattern “username@mydomain.com”`	Specifies a Cisco IOS regular expression (regex) pattern that matches the traffic pattern for the e-mail sender or user accounts from suspected domains that are causing the spam e-mail. In the example, “username” is configured as the name for a fake e-mail account used to discover senders (and their domain) when they try to send spam e-mail to this fake account.
Step 5	exit	Exits parameter-map profile configuration mode.

	Command or Action	Purpose
Step 6	**class-map type inspect smtp** *class-map-name* **Example:** `Router(config)# class-map type inspect smtp c1`	Creates a class map for the SMTP protocol and enters class-map configuration mode.
Step 7	**match recipient address regex** *parameter-map-name* **Example:** `Router(config-cmap)# match recipient address regex known-unknown-users`	Specifies the nonexistent e-mail recipient pattern in order to learn spam senders and their domain information by luring them to use this contrived e-mail recipient.
Step 8	exit	Exits class-map configuration mode.
Step 9	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.
Step 10	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures SMTP inspection parameters for this class map.
Step 11	**log** **Example:** `Router(config-pmap)# log`	Logs an action related to this class-type in the SMTP policy map.
Step 12	**reset** **Example:** `Router(config-pmap)# reset`	(Optional) Drops an SMTP connection with an SMTP sender (client) if it violates the specified policy. This action sends an error code to the sender and closes the connection gracefully.

Hiding Specified Private SMTP Commands on an SMTP Connection

Use this task to hide or “mask” commonly encountered SMTP verbs (SMTP commands) or specified private SMTP verbs used to provision an SMTP connection.

Specified verbs, such as the ATRN, ETRN, BDAT verbs may be considered vulnerable to exploitation if seen by a sender (client). The most commonly encountered SMTP verbs are listed along with the facility to specify a private verb as a string (using the WORD option).

Note The BDAT verb (used as an alternative to DATA) is not used, so in its place, the CHUNKING keyword is masked in the EHLO response. However, if the sender (client) continues to send the BDAT command, it is masked.

Note Using the **mask** command applies to certain **match** command filters like **match cmd verb**. Validations are performed to make this check and the configuration is not be accepted in case of invalid combinations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect smtp** *class-map-name*
4. **match cmd verb** {*verb-name* | *WORD*}
5. exit
6. **policy-map type inspect smtp** *policy-map-name*
7. **class type inspect smtp** *class-map-name*
8. **mask**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect smtp** *class-map-name* **Example:** `Router(config)# class-map type inspect smtp c1`	Creates a class map for the SMTP protocol and enters class-map configuration mode.

	Command or Action	Purpose

Step 4	**match cmd verb** {*verb-name* | *WORD*}
	Example:
	`Router(config-cmap)# match cmd verb ATRN`
		Specifies either the private verb name to "mask" that is used to provision an SMTP connection.
		• The *verb-name* argument is the name of an SNMP command verb.
		• The *WORD* argument is the name of a user-specified SMTP command verb, which is treated as an unknown verb and is masked regardless of whether the 'mask action is configured for the class or not.

Step 5	exit	Exits class-map configuration mode.
Step 6	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.
Step 7	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures SMTP inspection parameters for this class map.
Step 8	**mask** **Example:** `Router(config-pmap)# mask`	Explicitly masks the specified SMTP commands or the parameters returned by the server in response to an EHLO command.

Preventing a DoS Attack by Limiting the Length of the SMTP Header

A DoS attack (also called a buffer overflow attack) by a malicious sender (client) can cause the SMTP application firewall to lose time and memory while trying to reassemble the fake packets (large e-mail headers) associated with the e-mail. In an SMTP transaction, the header portion of an e-mail is considered part of the DATA area, which contains fields like Subject, From, To, Cc, Date, and proprietary information, which is used by a recipient's e-mail agent to process the e-mail. A DoS attack can be prevented by using the **match header length gt** command to limit the length of the SMTP header that can be received. If a match is found, possible actions that can be specified within the policy are as follows: allow, reset, or log (the log action triggers a syslog message when a match is found).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect smtp** *class-map-name*
4. **match header length gt** *bytes*
5. exit
6. **policy-map type inspect smtp** *policy-map-name*
7. **class type inspect smtp** *class-map-name*
8. **reset**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect smtp** *class-map-name* **Example:** `Router(config)# class-map type inspect smtp c1`	Creates a class map for the SMTP protocol and enters class-map configuration mode.
Step 4	**match header length gt** *bytes* **Example:** `Router(config-cmap)# match header length gt 16000`	Specifies a value from 1 to 65535 that limits the maximum length of the SMTP header in bytes to thwart DoS attacks.
Step 5	exit	Exits class-map configuration mode.
Step 6	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.

	Command or Action	Purpose
Step 7	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures SMTP inspection parameters for this class map.
Step 8	**reset** **Example:** `Router(config-pmap)# reset`	(Optional) Drops an SMTP connection with an SMTP sender (client) if it violates the specified policy. This action sends an error code to the sender and closes the connection gracefully.

Preventing a DoS Attack by Limiting the Length or TYPE of SMTP Command Line

The following task is used to limit the length of an SMTP command line to prevent a DoS attack, which occurs when a malicious sender (client) specifies large command lines in an e-mail to perform DoS attacks on SMTP servers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}*
4. **match cmd** {**line length gt** *length* | **verb** {**AUTH** | **DATA** | **EHLO** | **ETRN** | **EXPN** | **HELO** | **HELP** | **MAIL NOOP** | **QUIT** | **RCPT** | **RSET** | **SAML** | **SEND** | **SOML** | **STARTTLS** | **VERB** | **VRFY** | **WORD**}}
5. exit
6. **policy-map type inspect smtp** *policy-map-name*
7. **class type inspect smtp** *class-map-name*
8. **reset**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}* **Example:** `Router(config)# class-map type inspect smtp c1`	Enters class-map configuration mode and creates a class map for the SMTP protocol. • The *class-map-name* argument by itself specifies a single class-map. • The **match-all** keyword and *class-map-name* argument places logical and all matching statements under this class map. • The **match-any** keyword and *class-map-name* argument places logical or all matching statements under this class map. **Note** If no **match cmd verb** command statement is specified in a **class-map type inspect smtp match-all** command statement for a class-map, which contains the **match cmd line length gt** command statement, then the class-map applies to all SMTP commands.

Step 4	**match cmd** {**line length gt** *length* | **verb** {**AUTH** | **DATA** | **EHLO** | **ETRN** | **EXPN** | **HELO** | **HELP** | **MAIL NOOP** | **QUIT** | **RCPT** | **RSET** | **SAML** | **SEND** | **SOML** | **STARTTLS** | **VERB** | **VRFY** | **WORD**}}
	Example:
	`Router(config-cmap)# match header length gt 16000`
		Specifies a value that limits the length of the ESMTP command line or ESMTP command line verb used to thwart DoS attacks.
		• The *length* argument specifies the ESMTP command line greater than the length of a number of characters from 1 to 65535.

Step 5	exit	Exits class-map configuration mode.
Step 6	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.
Step 7	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures an SMTP class-map firewall for SMTP inspection parameters.

	Command or Action	Purpose
Step 8	**reset** **Example:** `Router(config-pmap)# reset`	(Optional) Drops an SMTP connection with an SMTP sender (client) if it violates the specified policy. This action sends an error code to the sender and closes the connection gracefully.

Examples

The following configuration has class-map c2 match when the length of the e-mail (MAIL) command exceeds 256 bytes.

When the **class-map type inspect smtp match-all** command statement is configured with the **match cmd verb** command statement, only the **match cmd line length gt** command statement can coexist.

```
class-map type inspect smtp match-all c2
  match cmd line length gt 256
  match cmd verb MAIL
```

There are no match restrictions in case of a **class-map type inspect smtp match-any** command statement for a class map because the class-map applies to all SMTP commands.

Restricting Content File Types in the Body of the E-Mail

The **match mime content-type regex** command is used to specify MIME content file types, which are restricted in attachments in the body of the e-mail being sent over SMTP. See the Example: MIME E-Mail Format section for more information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type regex** *parameter-map-name*
4. **pattern** *traffic-pattern*
5. exit
6. **class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}*
7. **match mime content-type regex** *content-type-regex*
8. exit
9. **policy-map type inspect smtp** *policy-map-name*
10. **class type inspect smtp** *class-map-name*
11. **log**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type regex** *parameter-map-name* **Example:** `Router(config)# parameter-map type regex jpeg`	Enter the parameter-map name of a specific traffic pattern. Once the parameter-map name is specified, parameter-map profile configuration mode is entered.
Step 4	**pattern** *traffic-pattern* **Example:** `Router(config-profile)# pattern “*image//*”`	Specifies a Cisco IOS regular expression (regex) pattern that matches the traffic pattern for the e-mail sender or user accounts from suspected domains that are causing the spam e-mail.
Step 5	exit	Exits parameter-map profile configuration mode.
Step 6	**class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}* **Example:** `Router(config)# class-map type inspect smtp c1`	Enters class-map configuration mode and creates a class map for the SMTP protocol. - The *class-map-name* argument by itself specifies a single class-map. - The **match-all** keyword and *class-map-name* argument places logical and all matching statements under this class map. - The **match-any** keyword and *class-map-name* argument places logical or all matching statements under this class map.

	Command or Action	Purpose
Step 7	**match mime content-type regex** *content-type-regex* **Example:** `Router(config-cmap)# match mime content-type regex jpeg`	Specifies the MIME content file type, which are restricted in attachments in the body of the e-mail being sent over SMTP. • The *content-type-regex* argument is the type of content in the MIME header in regular expression form. This example lets the user specify any form of JPEG image content to be restricted. **Note** The actual content of the MIME part is not checked to see if it matches with the declared content-type in the MIME header.
Step 8	exit	Exits class-map configuration mode.
Step 9	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.
Step 10	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures an SMTP class-map firewall for SMTP inspection parameters.
Step 11	**log** **Example:** `Router(config-pmap)# log`	Logs an action related to this class-type in the SMTP policy map.

Restricting Unknown Content Encoding Types from Being Transmitted

Unknown MIME content-encoding types or values can be restricted from being transmitted over SMTP by using one of the following parameters with the **match mime encoding**command.

These preconfigured content-transfer-encoding types act as a filter on the content-transfer-encoding field in the MIME header within the SMTP body. The uuencode encoding type is not recognized as a standard type by the MIME RFCs because many subtle differences exist in its various implementations. However, since it is used by some mail systems, the **x-uuencode** type is included in the preconfigured list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}*
4. **match mime encoding** {**unknown** | *WORD* | *encoding-type*}
5. exit
6. **policy-map type inspect smtp** *policy-map-name*
7. **class type inspect smtp** *class-map-name*
8. **log**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}* **Example:** `Router(config)# class-map type inspect smtp c1`	Enters class-map configuration mode and creates a class map for the SMTP protocol. • The *class-map-name* argument by itself specifies a single class-map. • The **match-all** keyword and *class-map-name* argument places logical and all matching statements under this class map. • The **match-any** keyword and *class-map-name* argument places logical or all matching statements under this class map.

	Command or Action	Purpose

Step 4	**match mime encoding** {**unknown** | *WORD* | *encoding-type*}
	Example:
	`Router (config-cmap)# match mime encoding quoted-printable`
		Restricts unknown MIME content-encoding types or values.
		• The **unknown** keyword is used if content-transfer-encoding value in the e-mail does not match any of the ones in the list to restrict unknown and potentially dangerous encodings.
		• The *WORD* argument is a user-defined content-transfer encoding type, which must begin with “X-” (for example, “X-myencoding-scheme”).
		• The *encoding-type* argument specifies one of the following preconfigured content-transfer-encoding types:
		◦ **7-bit**-ASCII characters
		◦ **8-bit**-Facilitates the exchange of e-mail messages containing octets outside the 7-bit ASCII range.
		◦ **base64**-Any similar encoding scheme that encodes binary data by treating it numerically and translating it into a base 64 representation.
		◦ **quoted-printable**-Encoding using printable characters (that is alphanumeric and the equals sign "=") to transmit 8-bit data over a 7-bit data path. It is defined as a MIME content transfer encoding for use in Internet e-mail.
		◦ **binary**-Representation for numbers using only two digits (usually, 0 and 1).
		◦ **x-uuencode**-Nonstandard encoding.
		Note The **quoted-printable** and **base64** encoding types tell the e-mail client that a binary-to-text encoding scheme was used and that appropriate initial decoding is necessary before the message can be read with its original encoding.

Step 5	exit	Exits class-map configuration mode.
Step 6	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.
Step 7	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures an SMTP class-map firewall for SMTP inspection parameters.

	Command or Action	Purpose
Step 8	**log** **Example:** `Router(config-pmap)# log`	Logs an action related to this class-type in the SMTP policy map.

Specifying a Text String to Be Matched and Restricted in the Body of an E-Mail

The **match body regex** command can be used to specify an arbitrary text expression to restrict specified content-types and content encoding types for text and HTML in the body of the e-mail. The text or HTML pattern is scanned only if the encoding is 7-bit or 8-bit and the encoding is checked before attempting to match the pattern. If the pattern is of another encoding type (for example, base64, zip files, and so on), then the pattern cannot be scanned.

Note Using this command can impact performance because the complete SMTP connection has to be scanned.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type regex** *parameter-map-name*
4. **pattern** *traffic-pattern*
5. exit
6. **class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}*
7. **match body regex** *parameter-map-name*
8. exit
9. **policy-map type inspect smtp** *policy-map-name*
10. **class type inspect smtp** *class-map-name*
11. **log**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type regex** *parameter-map-name* **Example:** `Router(config)# parameter-map type regex doc-data`	Enter the parameter-map name of a specific traffic pattern. Once the parameter-map name is specified, parameter-map profile configuration mode is entered.
Step 4	**pattern** *traffic-pattern* **Example:** `Router(config-profile)# pattern “*UD-421590*”`	Specifies a Cisco IOS regular expression (regex) pattern that matches the traffic pattern for the e-mail sender or user accounts from suspected domains that are causing the spam e-mail.
Step 5	exit	Exits parameter-map profile configuration mode.
Step 6	**class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}* **Example:** `Router(config)# class-map type inspect smtp c1`	Enters class-map configuration mode and creates a class map for the SMTP protocol. • The *class-map-name* argument by itself specifies a single class-map. • The **match-all** keyword and *class-map-name* argument places logical and all matching statements under this class map. • The **match-any** keyword and *class-map-name* argument places logical or all matching statements under this class map.
Step 7	**match body regex** *parameter-map-name* **Example:** `Router(config-cmap)# match body regex doc-data`	Specifies an arbitrary text expression to restrict specified content-types and content encoding types for text and HTML in the “body” of the e-mail.
Step 8	exit	Exits class-map configuration mode.
Step 9	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.

	Command or Action	Purpose
Step 10	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures an SMTP class-map firewall for SMTP inspection parameters.
Step 11	**log** **Example:** `Router(config-pmap)# log`	Logs an action related to this class-type in the SMTP policy map.

Configuring the Monitoring of Text Patterns in an SMTP E-Mail Subject Field

The **match header regex** command can be used specify an arbitrary text expression in the SMTP e-mail message header (Subject field) or e-mail body such as Subject, Received, To, or other private header fields to monitor text patterns.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type regex** *parameter-map-name*
4. **pattern** *traffic-pattern*
5. exit
6. **class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}*
7. **match header regex** *parameter-map-name*
8. exit
9. **policy-map type inspect smtp** *policy-map-name*
10. **class type inspect smtp** *class-map-name*
11. **reset**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type regex** *parameter-map-name* **Example:** `Router(config)# parameter-map type regex lottery-spam`	Enter the parameter-map name of a specific traffic pattern. Once the parameter-map name is specified, parameter-map profile configuration mode is entered.
Step 4	**pattern** *traffic-pattern* **Example:** `Router(config-profile)# pattern “Subject:*lottery*”`	Specifies a Cisco IOS regular expression (regex) pattern that matches the traffic pattern for the e-mail sender or user accounts from suspected domains that are causing the spam e-mail.
Step 5	exit	Exits parameter-map profile configuration mode.
Step 6	**class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}* **Example:** `Router(config)# class-map type inspect smtp c1`	Enters class-map configuration mode and creates a class map for the SMTP protocol. • The *class-map-name* argument by itself specifies a single class-map. • The **match-all** keyword and *class-map-name* argument places logical and all matching statements under this class map. • The **match-any** keyword and *class-map-name* argument places logical or all matching statements under this class map.
Step 7	**match header regex** *parameter-map-name* **Example:** `Router(config-cmap)# match header regex lottery-spam`	Specifies an arbitrary text expression in the SMTP e-mail message header to monitor text patterns.

	Command or Action	Purpose
Step 8	exit	Exits class-map configuration mode.
Step 9	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.
Step 10	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures an SMTP class-map firewall for SMTP inspection parameters.
Step 11	**reset** **Example:** `Router(config-pmap)# reset`	(Optional) Drops an SMTP connection with an SMTP sender (client) if it violates the specified policy. This action sends an error code to the sender and closes the connection gracefully.

Configuring a Parameter to Be Identified and Masked in the EHLO Server Reply

The **match reply ehlo** command is used to identify and mask a service extension parameter in the EHLO server reply (for example, 8BITMIME and ETRN) to prevent a sender (client) from using that particular service extension.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}*
4. **match reply ehlo** {*parameter* | *WORD*}
5. exit
6. **policy-map type inspect smtp** *policy-map-name*
7. **class type inspect smtp** *class-map-name*
8. **log**
9. **mask**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}* **Example:** `Router(config)# class-map type inspect smtp c1`	Enters class-map configuration mode and creates a class map for the SMTP protocol. - The *class-map-name* argument by itself specifies a single class-map. - The **match-all** keyword and *class-map-name* argument places logical and all matching statements under this class map. - The **match-any** keyword and *class-map-name* argument places logical or all matching statements under this class map.

Step 4	**match reply ehlo** {*parameter* | *WORD*}
	Example:
	`Router(config-cmap)# match reply ehlo ETRN`
		Identifies and masks a service extension parameter in the EHLO server reply.
		- The *parameter* argument specifies a parameter from the well-known EHLO keywords.
		- The *WORD* argument specifies an extension which is not on the EHLO list.

Step 5	exit	Exits class-map configuration mode.
Step 6	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.
Step 7	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures an SMTP class-map firewall for SMTP inspection parameters.

	Command or Action	Purpose
Step 8	**log** **Example:** `Router(config-pmap)# log`	Logs an action related to this class-type in the SMTP policy map.
Step 9	**mask** **Example:** `Router(config-pmap)# mask`	Explicitly masks the specified SMTP commands or the parameters returned by the server in response to an EHLO command.

Configuring a Logging Action for a Class Type in an SMTP Policy-Map

A logging action can be configured for a class type in an SMTP policy-map when conditions specified by the traffic class are met. The logging action results in a LOG_WARNING syslog message followed by the specific log message. The log message format is similar to other application firewall modules (for example, HTTP, IM, Peer-to-Peer (P2P)); session initiator/responder information, and zone-pair and class names.

Note The log action currently exists for other types of policy-maps (http, pop3).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}*
4. **match cmd verb** {*parameter* | *WORD*}
5. exit
6. **policy-map type inspect smtp** *policy-map-name*
7. **class type inspect smtp** *class-map-name*
8. **log**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect smtp** {*class-map-name* | **match-all** *class-map-name* | **match-any** *class-map-name}* **Example:** `Router(config)# class-map type inspect smtp c1`	Enters class-map configuration mode and creates a class map for the SMTP protocol. • The *class-map-name* argument by itself specifies a single class-map. • The **match-all** keyword and *class-map-name* argument places logical and all matching statements under this class map. • The **match-any** keyword and *class-map-name* argument places logical or all matching statements under this class map.

Step 4	**match cmd verb** {*parameter* | *WORD*}
	Example:
	`Router(config-cmap)# match cmd verb ATRN`
		Identifies and masks a service extension parameter in the EHLO server reply.
		• The *parameter* argument specifies a parameter from the well-known EHLO keywords.
		• The *WORD* argument specifies an extension which is not on the EHLO list.

Step 5	exit	Exits class-map configuration mode.
Step 6	**policy-map type inspect smtp** *policy-map-name* **Example:** `Router(config)# policy-map type inspect smtp p1`	Creates a Layer 7 SMTP policy map and enters policy-map configuration mode.
Step 7	**class type inspect smtp** *class-map-name* **Example:** `Router(config-pmap)# class type inspect smtp c1`	Configures an SMTP class-map firewall for SMTP inspection parameters.
Step 8	**log** **Example:** `Router(config-pmap)# log`	Logs an action related to this class-type in the SMTP policy map.

Configuration Examples for Application Inspection and Control for SMTP

Example Creating a Pinhole for the SMTP Port

The following example shows a configuration without any Layer 7 SMTP policy that creates a pinhole only for the SMTP port. Any command sent to the server, including the EHLO command is accepted.

```
class-map type inspect smtp c1
match protocol smtp
policy-map type inspect smtp c1
    class type inspect smtp c1
        inspect
```

Note No SMTP policy is configured by default. If an SMTP policy is not configured, then no SMTP inspection is done by default.

Example Preventing ESMTP Inspection

If a user decides to create a workable policy that is configured for SMTP inspection only, then it now needs to be explicitly specified in the policy.

The following example can be used to prevent ESMTP inspection:

```
class-map type inspect smtp c1
    match cmd verb EHLO
policy-map type inspect smtp c1
    class type inspect smtp c1
        mask
```

Example MIME E-Mail Format

The format of data being transmitted through SMTP is specified by using the MIME standard, which uses headers to specify the content-type, encoding, and the filenames of data being sent (text, html, images, applications, documents and so on). The following is an example of an e-mail using the MIME format:

```
From: "username2" <username2@example.com>
To: username3 <username3@example.com>
Subject: testmail
Date: Sat, 7 Jan 2006 20:18:47 -0400
Message-ID: <000dadf7453e$bee1bb00$8a22f340@oemcomputer>
MIME-Version: 1.0
Content-Type: image/jpeg;
name='picture.jpg'
Content-Transfer-Encoding: base64
```

<base64 encoded data for the picture.jpg image>

In the above example, the “name=’picture.jpg’” is optional. Even without the definition, the image is sent to the recipient. The e-mail client of the recipient may display the image as “part-1” or “attach-1” or it may render the image in-line. Also, attachments are not ‘stripped’ from the e-mail. If a content-type for which reset action was configured is detected, an 5XX error code is sent and the connection is closed, in order to prevent the whole e-mail from being delivered. However, the remainder of the e-mail message is sent.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Firewall commands	*Cisco IOS Security Command Reference*
ESMTP firewall information.	ESMTP Support for Cisco IOS Firewall
Information for configuring an SMTP policy.	Zone-Based Policy Firewall

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1869 and other SMTP RFC extensions apart from RFC 821.	SMTP Service Extensions

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 Application Inspection and Control for SMTP

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 Application Inspection and Control for SMTP

Feature Name	Releases	Feature Information
Application Inspection and Control for SMTP	12.4(20)T	The Application Inspection and Control for SMTP feature provides an intense provisioning mechanism that can be configured to inspect packets on a granular level so that malicious network activity, related to the transfer of e-mail at the application level, can be identified and controlled. This feature qualifies the Cisco IOS firewall extended SMTP (ESMTP) module as an “SMTP application firewall,” which protects in a similar way to that of an HTTP application firewall. The following commands were introduced or modified by this feature: **log (policy-map and class-map) , mask (policy-map)**, **match body regex**, **match cmd**, **match header length gt**, **match header regex**, **match mime content-type regex**, **match mime encoding**, **match sender address regex**, **match recipient address regex**, **match recipient count gt**, **match recipient invalid count gt**, **match reply ehlo**, **reset (policy-map)**.

Glossary

C3PL --Cisco Common Classification Policy Language. Structured, feature-specific configuration commands that use policy maps and class maps to create traffic policies based on events, conditions, and actions.

EHLO --Extended HELO substitute command for starting the capability negotiation. This command identifies the sender (client) connecting to the remote SMTP server by using the ESMTP protocol.

ESMTP --Extended Simple Mail Transfer Protocol. Extended version of the Simple Mail Transfer Protocol (SMTP), which includes additional functionality, such as delivery notification and session delivery. ESMTP is described in RFC 1869, SMTP Service Extensions.

HELO --Command that starts the SMTP capability negotiation. This command identifies the sender (client) connecting to the remote SMTP server by its fully qualified DNS hostname.

MAIL FROM --Start of an e-mail message that identifies the sender e-mail address (and name, if used), which appears in the From: field of the message.

MIME --Multipurpose Internet Mail Extension. Standard for transmitting nontext data (or data that cannot be represented in plain ASCII code) in e-mail, such as binary, foreign language text (such as Russian or Chinese), audio, or video data. MIME is defined in RFC 2045.

RCPT TO --Recipient e-mail address (and name, if used) that can be repeated multiple times for a likely message to deliver a single message to multiple recipients.

SMTP --Simple Mail Transfer Protocol. Internet protocol providing e-mail services.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.