parameter

To specify parameters for an enrollment profile, use the **parameter** command in ca-profile-enroll configuration mode. To disable specified parameters, use the **no** form of this command.

parameter *number* {**value** *value* | **prompt** *string*}

no parameter *number* {**value** *value* | **prompt** *string*}

Syntax Description

number	User parameters. Valid values range from 1 to 8.
value *value*	To be used if the parameter has a constant value.
prompt *string*	To be used if the parameter is supplied after the **crypto ca authenticate** command or the **crypto ca enroll** command has been entered. **Note** The value of the *string* argument does not have an effect on the value that is used by the router.

Defaults

No enrollment profile parameters are specified.

Command Modes

Ca-profile-enroll configuration

Command History

Release	Modification
12.2(13)ZH	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Usage Guidelines

The **parameter** command can be used within an enrollment profile after the **authentication command** command or the **enrollment command** has been enabled.

Examples

The following example shows how to specify parameters for the enrollment profile named “E”:

```
crypto ca trustpoint Entrust
  enrollment profile E
  serial

crypto ca profile enrollment E
 authentication url  http://entrust:81
 authentication command  GET /certs/cacert.der
 enrollment url  http://entrust:81/cda-cgi/clientcgi.exe
 enrollment command  POST reference_number=$P2&authcode=$P1
&retrievedAs=rawDER&action=getServerCert&pkcs10Request=$REQ
 parameter 1 value aaaa-bbbb-cccc
 parameter 2 value 5001
```

Related Commands

Command	Description
authentication command	Specifies the HTTP command that is sent to the CA for authentication.
crypto ca profile enrollment	Defines an enrollment profile.
enrollment command	Specifies the HTTP command that is sent to the CA for enrollment.

parameter-map type

To create or modify a parameter map, use the **parameter-map type** command in global configuration mode. To delete a parameter map from the configuration, use the **no** form of this command.

parameter-map type {**inspect** | **urlfilter** | **protocol-info** | **consent**} *parameter-map-name*

no parameter-map type {**inspect** | **urlfilter** | **protocol-info** | **consent**} *parameter-map-name*

Syntax Description

inspect	Defines an inspect type parameter map, which configures connection thresholds, timeouts, and other parameters pertaining to the **inspect** action.
urlfilter	Defines a URL-filter-specific parameter map.
protocol-info	Defines an application-specific parameter map. **Note** Protocol-specific parameter maps can be created only for Instant Messenger (IM) applications (AOL, I Seek You (ICQ), MSN Messenger, Yahoo Messenger and Windows Messenger).
consent	Defines an authentication proxy consent parameter map.
parameter-map-name	Name of the parameter map.

Command Default

None

Command Modes

Global configuration (config)

Command History

Release	Modification
12.4(6)T	This command was introduced.
12.4(9)T	The **protocol-info** keyword was added.
12.4(15)T	The **consent** keyword was added.
12.4(20)T	Support for ICQ and Windows Messenger was added.

Usage Guidelines

A parameter map allows you to specify parameters that control the behavior of actions and match criteria specified under a policy map and a class map, respectively.

There are currently four types of parameter maps:

- Inspect parameter map

  An inspect parameter map is optional. If you do not configure a parameter map, the software uses default parameters. Parameters associated with the inspect action apply to all nested actions (if any). If parameters are specified in both the top and lower levels, those in the lower levels override those in the top levels.

- URL filter parameter map

  A parameter map is required for URL filtering (via the URL filter action in a Layer 3 or Layer 4 policy map and the URL filter parameter map).

- Protocol-specific parameter map

  A parameter map is required for an IM application (Layer 7) policy map.
- Authentication proxy consent-specific parameter map.

Examples

The following example shows how to configure an IM-based firewall policy. In this example, all Yahoo Messenger and ICQ traffic is allowed to pass through, while all MSN Messenger, AOL and Windows Messenger traffic is blocked. Also, parameter maps are defined to control all Yahoo Messenger and ICQ traffic on a more granular level.

```
!
!
parameter-map type protocol-info ymsgr-servers
 server name messenger.yahoo.akadns.net
 server name .*.yahoo.com snoop
 server ip 192.0.2.100
 server ip range 192.0.2.115 192.0.2.180
parameter-map type protocol-info icq-servers
 server name login.oscar.aol.com
 server name .*.aol.com snoop
 server ip 192.0.2.200
 server ip range 192.0.2.215 192.0.2.230
!
!
class-map type inspect match-all l4-cmap-ymsgr
 match protocol ymsgr ymsgr-servers
class-map type inspect ymsgr match-any l7-cmap-ymsgr
 match  service text-chat
class-map type inspect match-all l4-cmap-icq
 match protocol icq icq-servers
class-map type inspect icq match-any l7-cmap-icq
 match  service text-chat
 match  service any
!
!
policy-map type inspect im l7-pmap-ymsgr
 class type inspect ymsgr l7-cmap-ymsgr
  allow
  log
policy-map type inspect im l7-pmap-icq
 class type inspect icq l7-cmap-icq
  allow
  log
policy-map type inspect to_internet
 class type inspect l4-cmap-ymsgr
  inspect
  service-policy im l7-pmap-ymsgr
 class type inspect l4-cmap-icq
  inspect
  service-policy im l7-pmap-icq
 class class-default
  drop
!
!
```

The following example shows a typical URL filter parameter map configuration:

```
parameter-map type urlfilter eng-filter-profile
 server vendor n2h2 172.16.1.2 port 3128 outside log timeout 10 retrans 6
 max-request 80
 max-resp-pak 200
 cache 200
 exclusive-domain permit cisco.com
 exclusive-domain deny gaming.com
```

The following example shows a sample inspect type parameter map configuration:

```
parameter-map type inspect eng_network_profile
 audit-trail on
 alert off
 max-incomplete low 2000
 max-incomplete high 3000
 one-minute low 5000
 one-minute high 8000
 udp idle-time 75
 dns-timeout 25
 tcp idle-time 90
 tcp finwait-time 20
 tcp synwait-time 10
 tcp block-non-session
 tcp max-incomplete host 2000 block-time 120
```

The following example shows how to define the consent-specific parameter map "consent_parameter_map" and a default consent parameter map:

```
parameter-map type consent consent_parameter_map
 copy tftp://192.168.104.136/consent_page.html flash:consent_page.html
 authorize accept identity consent_identity_policy
 timeout file download 35791
 file flash:consent_page.html
 logging enabled
 exit
!
parameter-map type consent default
 copy tftp://192.168.104.136/consent_page.html flash:consent_page.html
 authorize accept identity test_identity_policy
 timeout file download 35791
 file flash:consent_page.html
 logging enabled
 exit
!
```

parameter-map type inspect

To configure an inspect type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action, use the **parameter-map type inspect** command in global configuration mode. To delete an inspect type parameter map, use the **no** form of this command.

parameter-map type inspect {*parameter-map-name* | **global** | **default**}

no parameter-map type inspect {*parameter-map-name* | **global** | **default**}

Syntax Description

parameter-map-name	Name of the inspect parameter map.
global	Defines a global inspect parameter map.
default	Defines a default inspect parameter map.

Command Default

No inspect type parameter maps are set.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.4(6)T	This command was introduced.
15.1(1)T	The keywords **global** and **default** were added.

Usage Guidelines

After you enter the **parameter-map type inspect** command, you can enter the following commands in parameter-map type inspect configuration mode:

- **alert** {**on** | **off**}

  Turns on Cisco IOS stateful packet inspection alert messages.

- **audit-trail** {**on** | **off**}

  Turns audit trail messages on or off.

- **dns-timeout** *seconds*

  Specifies the Domain Name System (DNS) idle timeout.

- **icmp idle-timeout** *seconds*

  Configures the timeout for Internet Control Message Protocol (ICMP) sessions.

- **max-incomplete** {**low** | **high**} *number-of-connections*

  Defines the number of existing half-open sessions that will cause the software to start and stop deleting half-open sessions.

- **one-minute** {**low** | **high**} *number-of-connections*

  Defines the rate of new half-open session initiation in one minute that will cause the system to start deleting half-open sessions and stop deleting half-open sessions.

- **tcp finwait-time** *seconds*

  Specifies how long a TCP session will be managed after the Cisco IOS firewall detects a FIN-exchange.
- **tcp idle-time** *seconds*

  Configures the timeout for TCP sessions.
- **tcp max-incomplete host** *threshold* [**block-time** *minutes*}

  Specifies threshold and blocking time values for TCP host-specific denial-of-service (DOS) detection and prevention.
- **tcp synwait-time** *seconds*

  Specifies how long the software will wait for a TCP session to reach the established state before dropping the session.
- **udp idle-time** *seconds*

  Configures the timeout of User Datagram Protocol (UDP) sessions going through the firewall.

For more detailed information about these commands, see their individual command descriptions.

Examples

The following example shows a sample inspect parameter map with the Cisco IOS stateful packet inspection alert messages enabled:

```
parameter-map type inspect eng-network-profile
 alert on
```

The following example shows a sample inspect type parameter map configuration:

```
parameter-map type inspect eng_network_profile
 audit-trail on
 alert on
 max-incomplete low unlimited
 max-incomplete high unlimited
 one-minute low unlimited
 one-minute high unlimited
 udp idle-time 30
 icmp idle-time 10
 dns-timeout 5
 tcp idle-time 3600
 tcp finwait-time 5
 tcp synwait-time 30
 tcp block-non-session
 tcp max-incomplete host 1-2147483647 block-time unlimited
 sessions maximum:2147483647
```

Related Commands

Command	Description
alert	Turns on Cisco IOS stateful packet inspection alert messages.
audit-trail	Turns audit trail messages on and off.
dns-timeout	Specifies the DNS idle timeout.
icmp idle-timeout	Configures the timeout for ICMP sessions.
inspect	Enables Cisco IOS stateful packet inspection.

Command	Description
max-incomplete	Defines the number of existing half-open sessions that will cause the software to start and stop deleting half-open sessions.
one-minute	Defines the number of new unestablished sessions that will cause the system to start deleting half-open sessions and stop deleting half-open sessions.
tcp finwait-time	Specifies how long a TCP session will be managed after the Cisco IOS firewall detects a FIN-exchange.
tcp idle-time	Configures the timeout for TCP sessions.
tcp max-incomplete host	Specifies threshold and blocking time values for TCP host-specific denial-of-service (DOS) detection and prevention.
tcp synwait-time	Specifies how long the software will wait for a TCP session to reach the established state before dropping the session.
udp idle-time	Configures the timeout of UDP sessions going through the firewall.

parameter-map type protocol-info

To create or modify a protocol-specific parameter map and enter parameter-map type configuration mode, use the **parameter-map type protocol-info** command in global configuration mode. To delete a protocol-specific parameter map from the configuration, use the **no** form of this command.

parameter-map type protocol-info [**msrpc** | **sip** | **stun-ice**] *parameter-map-name*

no parameter-map type protocol-info [**msrpc** | **sip** | **stun-ice**] *parameter-map-name*

Syntax Description

msrpc	(Optional) Defines a Microsoft Remote Procedure Call (MSRPC) protocol-info parameter map.
sip	(Optional) Defines a Session Initiation Protocol (SIP) protocol-info parameter map.
stun-ice	(Optional) Defines a Session Traversal Utilities for Network Address Translation (NAT) and Interactive Connectivity Establishment (STUN-ICE) protocol-info parameter map.
parameter-map-name	Name of the parameter map.

Command Default

No protocol-specific parameter maps are created.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.4(11)T	This command was introduced.
15.0(1)M	This command was modified. The **sip** keyword was added.
15.1(4)M	This command was modified. The **msrpc** keyword was added.

Usage Guidelines

A protocol-specific parameter map allows you to specify the parameters that control the behavior of actions specified under a policy map and match criteria specified under a class map.

Protocol-specific parameter maps can be created for real-time voice, video, and text messaging applications (such as AOL, MSN Messenger, or Windows Messenger).

Examples

The following example shows a sample SIP protocol type parameter map configuration. In this example, the parameter map is configured to not open a media channel when attached to a SIP class map:

```
Router(config)# parameter-map type protocol-info sip pmap-sip
Router(config-profile)# disable open-media channel
```

The following example shows a sample STUN-ICE protocol type parameter map configuration. In this example, the parameter map is configured to not open a media channel when attached to a SIP class map:

```
Router(config)# parameter-map type protocol-info stun-ice
Router(config-profile)# disable open-media channel
Router(config-profile)# authorization agent-id 20 shared-secret 12345flower12345
cat-window 15
```

The following example shows how to configure an Instant Messaging-based firewall policy. In this example, all Yahoo Messenger and I Seek You (ICQ) traffic is allowed to pass through, while all MSN Messenger, AOL, and Windows Messenger traffic is blocked. Also, parameter maps are defined to control all Yahoo Messenger and ICQ traffic on a more granular level.

```
Router(config)# parameter-map type protocol-info ymsgr-servers
Router(config-profile)# server name messenger.yahoo.akadns.net
Router(config-profile)# server name .*.yahoo.com snoop
Router(config-profile)# server ip 192.0.2.100
Router(config-profile)# server ip range 192.0.2.115 192.0.2.180
Router(config-profile)# exit

Router(config)# parameter-map type protocol-info icq-servers
Router(config-profile)# server name login.oscar.aol.com
Router(config-profile)# server name .*.aol.com snoop
Router(config-profile)# server ip 192.0.2.200
Router(config-profile)# server ip range 192.0.2.215 192.0.2.230
Router(config-profile)# exit

Router(config)# class-map type inspect match-all l4-cmap-ymsgr
Router(config-cmap)# match protocol ymsgr ymsgr-servers
Router(config-cmap)# exit

Router(config)# class-map type inspect ymsgr match-any l7-cmap-ymsgr
Router(config-cmap)# match  service text-chat
Router(config-cmap)# exit

Router(config)# class-map type inspect match-all l4-cmap-icq
Router(config-cmap)# match protocol icq icq-servers
Router(config-cmap)# exit

Router(config)# class-map type inspect icq match-any l7-cmap-icq
Router(config-cmap)# match  service text-chat
Router(config-cmap)# match  service any
Router(config-cmap)# exit

Router(config)# policy-map type inspect im l7-pmap-ymsgr
Router(config-pmap)# class type inspect ymsgr l7-cmap-ymsgr
Router(config-pmap-c)# allow
Router(config-pmap-c)# log
Router(config-pmap-c)# exit

Router(config)# policy-map type inspect im l7-pmap-icq
Router(config-pmap)#class type inspect icq l7-cmap-icq
Router(config-pmap-c)# allow
Router(config-pmap-c)# log
Router(config-pmap-c)# exit

Router(config)# policy-map type inspect to_internet
Router(config-pmap)# class type inspect l4-cmap-ymsgr
Router(config-pmap-c)# inspect
Router(config-pmap-c)# service-policy im l7-pmap-ymsgr
Router(config-pmap-c)# exit

Router(config-pmap)# class type inspect l4-cmap-icq
Router(config-pmap-c)# inspect
Router(config-pmap-c)# service-policy im l7-pmap-icq
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# drop
```

Related Commands

Command	Description
disable open-media-channel	Prevents the creation of RTP or RTCP media channels when a SIP class map is used for SIP inspection.
parameter-map type inspect	Configures an inspect type parameter map for connecting thresholds, timeouts, and other parameters pertaining to the inspect action.

parameter-map type inspect-vrf

To configure an inspect VPN Routing and Forwarding (VRF)-type parameter map, use the **parameter-map type inspect-vrf** command in global configuration mode. To delete an inspect VRF type parameter map, use the **no** form of this command.

parameter-map type inspect-vrf *vrf-pmap-name*

no parameter-map type inspect-vrf *vrf-pmap-name*

Syntax Description

vrf-pmap-name	Name of the parameter map.

Command Default

An inspect VRF-type parameter map is not configured.

Command Modes

Global configuration (config)

Command History

Release	Modification
Cisco IOS XE Release 3.3S	This command was introduced.

Examples

The following example shows how to configure an inspect VRF-type parameter map named inspect-pmap:

```
Router(config)# parameter-map type inspect-vrf inspect-pmap
```

Related Commands

Command	Description
parameter-map type	Creates or modifies a parameter map.
show parameter-map type inspect-vrf	Displays information about the configured inspect VRF-type parameter maps.

parameter-map type inspect-zone

To configure an inspect zone-type parameter map, use the **parameter-map type inspect-zone** command in global configuration mode. To remove an inspect zone type parameter map, use the **no** form of this command.

parameter-map type inspect-zone *zone-pmap-name*

no parameter-map type inspect-zone *zone-pmap-name*

Syntax Description

zone-pmap-name	Name of the parameter map.

Command Default

Inspect zone-type parameter maps are not configured.

Command Modes

Global configuration (config)

Command History

Release	Modification
Cisco IOS XE Release 3.3S	This command was introduced.

Examples

The following example shows how to create an inspect zone-type parameter map named zone-pmap:

```
Router(config)# parameter-map type inspect-zone zone-pmap
```

Related Commands

Command	Description
parameter-map type	Creates or modifies a parameter map.
show parameter-map type inspect-zone	Displays information about the configured inspect zone-type parameter maps.

parameter-map type regex

To configure a parameter-map type to match a specific traffic pattern, use the **parameter-map type regex** command in global configuration mode. To delete a parameter-map type with a regular expression (regex), use the **no** form of this command.

parameter-map type regex *parameter-map-name*

no parameter-map type regex

Syntax Description

parameter-map-name	Name of the parameter map. The name can be a maximum of 228 alphanumeric characters. **Note** Using blank spaces is not recommended. The system interprets the first blank space as the end of the parameter-map name unless the string contains blank spaces and is delimited by quotation marks.

Command Default

A regex parameter map is not configured.

Command Modes

Global configuration

Command History

Release	Modification
12.4(9)T	This command was introduced.

Usage Guidelines

You can enter a regex to match text strings either literally as an exact string or by using metacharacters so that you can match multiple variants of a text string. You can use a regex to match the content of certain application traffic; for example, you can match a uniform resource identifier (URI) string inside an HTTP packet using the **match request regex** command under an HTTP inspection class map.

Use Ctrl-V to ignore all of the special characters in the command line interface (CLI), such as a question mark (?) or a tab. For example, type **d**[**Ctrl-V**]**g** to enter **d?g** in the configuration.

Table 44 lists the metacharacters that have special meanings.

Table 44 regex Metacharacters

Character	Description	Notes
.	Dot	Matches any single character. For example, **d.g** matches dog, dag, dtg, and any word that contains those characters.
(*xxx*)	Subexpression	A subexpression segregates characters from surrounding characters, so that you can use other metacharacters on the subexpression. For example, **d(o|a)g** matches dog and dag, but **do|ag** matches do and ag. A subexpression can also be used with repeat quantifiers to differentiate the characters meant for repetition. For example, **ab(xy){3}z** matches abxyxyxyz.
|	Alternation	Matches either expression that it separates. For example, **dog|cat** matches dog or cat.
?	Question mark	A quantifier that indicates that there are 0 or 1 of the previous expression. For example, **lo?se** matches lse or lose. **Note** You must enter Ctrl-V and then the question mark or else the help function is invoked.
*	Asterisk	A quantifier that indicates that there are 0, 1 or any number of the previous expression. For example, **lo*se** matches lse, lose, loose, and so on.
+	Plus	A quantifier that indicates there is at least one occurrence of the previous expression. For example, **lo+se** matches lose and loose, but not lse.
{*x*}	Repeat quantifier	Repeat exactly *x* times. For example, **ab**(**xy**){**3**}**z** matches abxyxyxyz.
{*x*,}	Minimum repeat quantifier	Repeat at least *x* times. For example, **ab**(**xy**){**2,**}**z** matches abxyxyz, abxyxyxyz, and so on.
[*abc*]	Character class	Matches any character in the brackets. For example, [**abc**] matches a, b, or c.
[^*abc*]	Negated character class	Matches a single character that is not contained within the brackets. For example, [**^abc**] matches any character other than a, b, or c; and [**^A-Z**] matches any single character that is not an uppercase letter.
[*a*-*c*]	Character range class	Matches any character in the range. [**a-z**] matches any lowercase letter. You can mix characters and ranges; for example, [**abcq-z**] matches a, b, c, q, r, s, t, u, v, w, x, y, z, and so does [**a-cq-z**]. **Note** The dash (-) character is literal only if it is the last or the first character within the brackets, [**abc-**] or [**-abc**].
“ ”	Quotation marks	Preserves trailing or leading spaces in the string. For example, **“test”** preserves the leading space when it looks for a match.

Table 44 regex Metacharacters (continued)

Character	Description	Notes
^	Caret	Specifies the beginning of a line.
\	Escape character	When preceding a literal character, matches a literal character. For example, \[matches the left square bracket.
char	Character	When character is not a metacharacter, matches the literal character.
\r	Carriage return	Matches a carriage return 0x0d.
\n	New line	Matches a new line 0x0a.
\t	Tab	Matches a tab 0x09.
\f	Formfeed	Matches a form feed 0x0c.
\x*nn*	Escaped hexadecimal number	Matches an ASCII character using hexadecimal numbers (exactly two digits).
\nnn	Escaped octal number	Matches an ASCII character as an octal number (exactly three digits). For example, the character 040 represents a space.

Examples

The following example configures and applies a regex parameter map to an HTTP application firewall parameter-map type whose URI matches any of the following regular expressions:

- ".*cmd.exe"
- ".*money"
- ".*shopping"

```
Router# configure terminal
Router(config)# parameter-map type regex uri-regex-cm
Router(config-profile)# pattern ".*cmd.exe"
Router(config-profile)# pattern ".*money"
Router(config-profile)# pattern ".*shopping"
Router(config-profile)# exit

Router(config)# class-map type inspect http uri-check-cm
Router(config-cmap)# match request uri regex uri-regex-cm
Router(config-cmap)# exit

Router(config)# policy-map type inspect http uri-check-pm
Router(config-pmap)# class type inspect http uri-check-cm
Router(config-pmap-c)# reset
```

The following example configures a regex parameter map whose case-insensitive pattern matches multiple variants of the string "hello":

```
Router# configure terminal
Router(config)# parameter-map type regex body_regex
Router(config-profile)# pattern ".*[Hh][Ee][Ll][Ll][Oo]"
Router(config-profile)# end
```

Related Commands

Command	Description
class-map type inspect	Creates a Layer 3 and Layer 4 or a Layer 7 (application-specific) inspect type class map.
class type inspect	Specifies the traffic (class) on which an action is to be performed.
match request regex	Configures an HTTP firewall policy to permit or deny HTTP traffic on the basis of request messages whose URI or arguments (parameters) match a defined regular expression.
parameter-map type	Creates or modifies a parameter map.
policy-map type inspect	Creates a Layer 3 and Layer 4 or a Layer 7 (application-specific) inspect type policy map.

parameter-map type trend-global

To create or modify the parameter map for global parameters associated with a Trend Router Provisioning Server (TRPS) and to place the system in parameter map configuration mode, use the **parameter-map type trend-global** command in global configuration mode. To delete the global parameters associated with a TRPS from the configuration, use the **no** form of this command.

parameter-map type trend-global *parameter-map-name*

no parameter-map type trend-global *parameter-map-name*

Syntax Description

parameter-map-name	Name of the parameter map for the global parameters associated with the TRPS.

Command Default

No parameter map for the global TRPS parameters is created.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.4(15)XZ	This command was introduced.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.
15.1(2)T	This command was modified. The **pipeline**, **on**, and **off** keywords were added.

Usage Guidelines

Use the **parameter-map type trend-global** command to specify global parameters for the TRPS. You can specify only one trend-global parameter map on the system. To specify per-policy parameters, use the **parameter-map type urlfpolicy** command.

When you create or modify a global TRPS parameter map, use the following commands in parameter map configuration mode to set the values for the global TRPS parameters:

- **alert** {**on** | **off**}—Turns on or off URL-filtering server alert messages that are displayed on the console. The default is **on**.
- **cache-entry-lifetime** *hours*—Specifies how long, in hours, an entry remains in the cache table. Cache entries remain in the table until the cache-entry-lifetime value for the entry expires or until the cache is full, whichever occurs first. When the cache is full, the entry is removed to make room for subsequent entries. The range is from 1 to 120. The default is 24.
- **cache-size maximum-memory** *kilobyte*—Specifies the maximum size of the categorization cache, in kilobytes. The range is from 0 to 128000. The default is 256.
- **exit**—Exits from the parameter map.
- **no**—Negates or sets default values for a command.

- **server** {*server-name* | *ip-address*} [**http-port** *port-number*] [**https-port** *port-number*] [**retrans** *retransmission-count*] [**timeout** *seconds*] [**pipeline** {**on** | **off**}]—Specifies information about the TRPS. Use the server command in profile configuration mode.
  - **http-port** *port-number*—Specifies the HTTP port that is listening for requests. The range is from 1 to 65535. The default is 80.
  - **https-port** *port-number*—Specifies the HTTPS port that is listening for secure HTTP requests. The range is from 1 to 65535. The default is 443.
  - **pipeline** {**on** | **off**}—Turns on or off the TRPS pipeline requests. The default is **on**.
  - **retrans** *retransmission-count*—Specifies the number of times the router retransmits the lookup request when a response is not received from the TRPS. The range is from 1 to 5. The default is 3.
  - **server** {*server-name* | *ip-address*}—Specifies the domain name or the IP address of the server. The default is trps.trendmicro.com.
  - **timeout** *seconds*—Specifies the number of seconds that the router waits for a response from the TRPS. The range is from 1 to 300. The default is 60.

Examples

The following shows an example of how to specify global TRPS parameters in a parameter map named global-parameter-map:

```
parameter-map type trend-global global-parameter-map
 server server.example.com retrans 5 timeout 200
 cache-size maximum-memory 128000
 cache-entry-lifetime 1
```

Related Commands

Command	Description
alert	Turns on or off URL-filtering system alert messages that are displayed on the console.
cache-entry lifetime	Specifies how long an entry remains in the cache table.
cache-size maximum-memory	Specifies the size of the categorization cache.
parameter-map type urlfpolicy	Specifies per-policy URL filtering parameters.
server	Specifies information about the TRPS.

parameter-map type urlfilter

Note This command is hidden in releases later than Cisco IOS Release 12.4(20)T, but it continues to work. The **parameter-map type urlfpolicy** command can also be used. This command is used to create URL filtering parameters for local, trend, Websense Internet filtering, and the N2H2 Internet blocking program. We recommend the use of the URL filter policy rather than the URL filter action for Cisco IOS Release 12.4(20)T. All the use-cases supported by URL filter as an action are also supported by URL filter policy.

To create or modify a parameter map for URL filtering parameters, use the **parameter-map type urlfilter** command in global configuration mode. To delete a URL filter parameter map, use the **no** form of this command.

parameter-map type urlfilter *parameter-map-name*

no parameter-map type urlfilter *parameter-map-name*

Syntax Description

parameter-map-name	Name of the URL parameter map.

Command Default

None

Command Modes

Global configuration (config)

Command History

Release	Modification
12.4(6)T	This command was introduced.
12.4(15)XZ	This command was removed.

Usage Guidelines

When you are creating or modifying a URL parameter map, you can enter the following subcommands after you enter the **parameter-map type urlfilter** command. For more detailed information about the subcommands, see their individual command descriptions by going to the “Command Reference” section on page 45.

- **alert** {**on** | **off**}

  Turns on or off URL-filtering system alert messages that are displayed on the console.

- **allow-mode** {**on** | **off**}

  Turns on or off the default mode (allow mode) of the filtering algorithm.

- **audit-trail** {**on** | **off**}

  Turns on or off the logging of URL information into the syslog server or router.

- **cache** *number-of-entries*

  Configures cache parameters.

- **exclusive-domain** {**deny** | **permit**} *domain-name*

  Adds or removes a domain name to or from the exclusive domain list so that the Cisco IOS firewall does not have to send lookup requests to the vendor server.
- **max-request** *number-of-requests*

  Specifies the maximum number of outstanding requests that can exist at any given time.
- **max-resp-pak** *number-of-responses*

  Specifies the maximum number of HTTP responses that the Cisco IOS firewall can keep in its packet buffer.
- **server vendor** {**n2h2** | **websense**} {*ip-address* | *hostname* [**port** *port-number*]} [**outside**] [**log**] [**retrans** *retransmission-count*] [**timeout** *seconds*]

  Specifies a vendor server for URL filtering.
- **source-interface** *interface-name*

  Specifies the interface whose IP address will be used as the source IP address while making a TCP connection to the URL filter server (websense or N2h2).

Examples

The following example shows a sample URL parameter map:

```
parameter-map type urlfilter eng-network-profile
 server vendor n2h2 10.64.64.22 port 4128 outside retrans 4 timeout 8
```

The following example shows a typical URL filter configuration:

```
parameter-map type urlfilter eng-network-profile
 server vendor n2h2 10.64.65.22 port 3128 outside log retrans 6 timeout 10
 max-request 80
 max-resp-pak 200
 cache 200
 exclusive-domain permit cisco.com
 exclusive-domain deny gaming.com
```

Related Commands

Command	Description
alert	Turns on or off URL-filtering system alert messages that are displayed on the console.
allow-mode	Turns on or off the default mode (allow mode) of the filtering algorithm.
audit-trail	Turns on or off the logging of URL information into the syslog server or router.
cache	Configures cache parameters.
exclusive-domain	Adds or removes a domain name to or from the exclusive domain list so that the Cisco IOS firewall does not have to send lookup requests to the vendor server.
max-request	Specifies the maximum number of outstanding requests that can exist at any given time.
max-resp-pak	Specifies the maximum number of HTTP responses that the Cisco IOS firewall can keep in its packet buffer.
server vendor	Specifies a vendor server for URL filtering.

parameter-map type urlfpolicy

To create or modify a parameter map for a URL filtering policy and to place the system in parameter map configuration mode, use the **parameter-map type urlfpolicy** command in global configuration mode. To delete the parameter map for a URL filtering policy from the configuration, use the **no** form of this command.

parameter-map type urlfpolicy {**local** | **trend** | **n2h2** | **websense**} *parameter-map-name*

no parameter-map type urlfpolicy {**local** | **trend** | **n2h2** | **websense**} *parameter-map-name*

Syntax Description

local	Specifies that the parameters are for a local URL filtering policy. See Table 45 for more information.
trend	Specifies that the parameters are for a Trend Micro URL filtering policy. See Table 46 for more information.
n2h2	Specifies that the parameters are for a SmartFilter (previously N2H2) URL filtering policy. See Table 47 for more information.
websense	Specifies that the parameters are for a Websense URL filtering policy. See Table 47 for more information.
parameter-map-name	The name of the parameter map for a URL filtering policy.

Command Default

No parameter maps for a URL filtering policy are created.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.4(15)XZ	This command was introduced.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.

Usage Guidelines

Use the **parameter-map type urlfpolicy** command to create a parameter map for a URL filtering policy. The commands that you use to specify the parameters for a filtering policy depend on the URL filtering server you are using.

Table 45 defines the parameters for a local URL filtering policy.

Table 46 defines the per-policy parameters for a Trend Micro URL filtering policy. These parameters are in addition to the global Trend Micro policy parameters specified with the **parameter-map type trend-global** command.

Table 47 defines the per-policy parameters for SmartFilter (N2H2) and Websense URL filtering policies.

Table 45 Parameters for Local URL Filtering Policies

Syntax	Description
alert {**on** | **off**}	Turns on or off URL filtering alert messages that are displayed on the console. The default is **off**.
allow-mode {**on** | **off**}	Specifies whether to allow or block URL requests when the URL filtering process does not have connectivity to a URL filtering database. When **allow-mode** is **on**, all unmatched URL requests are allowed; when **off**, all unmatched URL requests are blocked. The default is **off**.
block-page {**message** *string* | **redirect-url** *url*}	Specifies the response to a blocked URL request. • **message** *string*—Specifies the message text to be displayed when a URL request is blocked. • **redirect-url** *url*—Specifies the URL of the web page to be displayed when a URL request is blocked.
exit	Exits from the parameter map.
no	Negates or sets default values for a command.

Table 46 Parameters for Trend Micro URL Filtering Policies

Syntax	Description
allow-mode {**on** | **off**}	Specifies whether to allow or block URL requests when the URL filtering process does not have connectivity to a URL filtering database. When **allow-mode** is **on**, all unmatched URL requests are allowed; when **off**, all unmatched URL requests are blocked. The default is **off**.
block-page {**message** *string* | **redirect-url** *url*}	Specifies the response to a blocked URL request. • **message** *string*—Specifies the message text to be displayed when a URL request is blocked. • **redirect-url** *url*—Specifies the URL of the web page to be displayed when a URL request is blocked.
exit	Exits from the parameter map.
max-request *number-requests*	Specifies the maximum number of pending requests. The range is from 1 to 2147483647. The default is 1000.
max-resp-pak *number-responses*	Specifies the number of HTTP responses that can be buffered. The range is from 0 and 20000. The default is 200.
no	Negates or sets default values for a command.
truncate hostname	Specifies that URLs be truncated at the end of the domain name.

Table 47 Parameters for SmartFilter and Websense URL Filtering Policies

Syntax	Description
alert {**on** | **off**}	Turns on or off URL filtering alert messages that are displayed on the console. The default is **off**.
allow-mode {**on** | **off**}	Specifies whether to allow or block URL requests when the URL filtering process does not have connectivity to a URL filtering database. When **allow-mode** is **on**, all unmatched URL requests are allowed; when **off**, all unmatched URL requests are blocked. The default is **off**.
block-page {**message** *string* | **redirect-url** *url*}	Specifies the response to a blocked URL request. • **message** *string*—Specifies the message text to be displayed when a URL request is blocked. • **redirect-url** *url*—Specifies the URL of the web page to be displayed when a URL request is blocked.
cache-entry-lifetime *hours*	Specifies how long, in hours, an entry remains in the cache table. The default is 24.
cache-size maximum-entries *number-entries*	Specifies the maximum number of entries that can be stored in the categorization cache. The default is 5000.
exit	Exits from the parameter map.
max-request *number-requests*	Specifies the maximum number of pending requests. The range is from 1 to 2147483647. The default is 1000.
max-resp-pak *number-responses*	Specifies the number of HTTP responses that can be buffered. The range is from 0 and 20000. The default is 200.
no	Negates or sets default values for a command.
server {*server-name* | *ip-address*} [**outside**] [**port** *port-number*] [**retrans** *retransmission-count*] [**timeout** *seconds*]	Specifies the parameters for the URL filtering server. • **server** {*server-name* | *ip-address*} Specifies the domain name or the IP address of the URL filtering server. • **outside** Specifies whether the URL filtering server is outside the network. • **port** *port-number* Specifies the port that is listening for requests. The range is from 1 to 65535. The default is 80. • **retrans** *retransmission-count* Specifies the number of times the router retransmits the lookup request when a response is not received from the Trend Router Provisioning Server (TRPS). The range is from 1 to 5. The default is 3. • **timeout** *seconds* Specifies the number of seconds that the router waits for a response from the TRPS. The range is from 1 to 300. The default is 60.

Table 47 Parameters for SmartFilter and Websense URL Filtering Policies (continued)

Syntax	Description
source-interface *interface-name*	Specifies the interface whose IP address will be used as the source IP address when a TCP connection is established between the system and the URL filtering server.
truncate {hostname | script-options}	Specifies that URLs be truncated. • **hostname** Specifies that URLs be truncated at the end of the domain name. • **script-options** Specifies that URLs be truncated at the left-most question mark in the URL.
urlf-server-log {on | off}	Enables sending information about HTTP requests to the URL filtering server's log server. The information includes the URL, the hostname, the source IP address, and the destination IP address.

Examples

The following example shows a parameter map for a local URL filtering policy that does not send alert messages and displays the message "URL is blocked by local filters" when a URL is blocked:

```
parameter-map type urlfpolicy local local-parameter-map
 alert off
 block-page message "URL is blocked by local-filters"
```

The following example shows a configuration for global parameters and per-policy parameters for a Trend Micro URL filtering policy:

```
parameter-map type trend-global global-parameter-map
 server mytrps.trendmicro.com retrans 5 timeout 200
 cache-size maximum-memory 128000
 cache-entry-lifetime 1

parameter-map type urlfpolicy trend trend-parameter-map
 max-request 2147483647
 max-resp-pak 20000
 truncate hostname
 block-page message "group2 is blocked by trend"
```

The following example shows the configuration for per-policy parameters for a SmartFilter URL filtering policy:

```
parameter-map type urlfpolicy n2h2 n2h2-parameter-map
 server n2h2Server timeout 30
 max-request 2000
 max-resp-pak 2000
 source-interface Loopback0
 truncate script-parameters
 cache-size maximum-entries 100
 cache-entry-lifetime 1
 block-page redirect-url http://www.example.com
```

Related Commands

Command	Description
parameter-map type trend-global	Specifies the global parameters associated with Trend Micro URL filtering policies.

parameter-map type urlf-glob

To create or modify a parameter map used to specify a list of domains, URL keywords, or URL metacharacters that should be allowed or blocked by local URL filtering, use the **parameter-map type urlf-glob** command in global configuration mode. To delete the parameter map, use the **no** form of this command.

parameter-map type urlf-glob *parameter-map-name*

no parameter-map type urlf-glob *parameter-map-name*

Syntax Description

parameter-map-name	Name of the parameter map for a local URL filtering policy.

Command Default

No URL filtering parameter maps are created.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.4(15)XZ	This command was introduced.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.

Usage Guidelines

The **parameter-map type urlf-glob** command can be used to create a parameter map for trusted domains, a parameter map for untrusted domains, and a parameter map for URL keywords. The following sub-commands are available in parameter map configuration mode to specify matching parameters when the **parameter-map type urlf-glob** command is issued:

- **exit**—Exits from URL filtering parameter map configuration mode.
- **no**—Negates or sets default values for a command.
- **pattern** *expression*—Configures a matching pattern that refers to a domain name, URL keyword, URL metacharacter entry, or URL keyword and URL metacharacter combination. The characters /, {, and } are not allowed in the expression. The question mark (?) is not allowed because it is reserved for the help function in the command-line interface (CLI).

URL pattern matching is improved because the period (.) is interpreted as a dot, and not as a wildcard entry representing a single character, as is the case with regex regular expression pattern matching.

A URL keyword is a complete word that occurs after the domain name and that is between the forward slash (/) path delimiters. For example in the URL http://www.example.com/hack/123.html, only “hack” and “123.html” are treated as keywords. Anything in the host or domain name can be allowed or blocked using a domain name, and thus a URL keyword should be a word that comes after the domain name. The entire keyword in the URL must match the pattern. For example if you have **pattern** *hack*, the URL www.example.com/hacksite/123.html doesn't match the pattern. In order to match this URL, you must have *hacksite*.

URL metacharacters allow pattern matching of single characters or ranges of characters to URLs, similar to the way a UNIX style glob expression works. The URL metacharacters are presented in Table 48.

Table 48 URL Metacharacters for URL Pattern Matching

Character	Description
*	Asterisk—matches any sequence of 0 or more characters.
[*abc*]	Character class—matches any character in the brackets. The character matching is case sensitive. For example, [abc] matches a, b, or c.
[*a-c*]	Character range class. Matches any character in the range. The character matching is case sensitive. [a-z] matches any lowercase letter. You can mix characters and ranges; for example, [abcq-z] matches a, b, c, q, r, s, t, u, v, w, x, y, z, and so does [a-cq-z]. **Note** The dash (-) character is literal only if it is the last or the first character within the brackets, [abc-] or [-abc].
[*0-9*]	Numerical range class. Matches any number in the brackets. For example [0-9] matches 0, 1, 2, 3, 4, 5, 6, 7, 8, or 9.

URL metacharacters are combined with domain names and URL keywords for pattern matching. For example, **pattern** *.example.com will match the domain name www.example.com and **pattern** www.[ey]xample.com can be used to block both www.example.com and www.yxample.com. Also, **pattern** www.example[0-9][0-9].com can be used to block www.example01.com, www.example33.com, and www.example99.com. An example of combining a keyword and metacharacter for pattern matching is using **pattern** hack* to block www.example.com/hacksite/123.html.

Examples

The following shows an example of specifying the parameter map for trusted domains:

```
Router(config)# parameter-map type urlf-glob trusted-domain-param
Router(config-profile)# pattern www.example.com
Router(config-profile)# pattern *.example2.com
```

The following shows an example of a parameter map specifying keywords to be blocked:

```
Router(config)# parameter-map type urlf-glob keyword-param
Router(config-profile)# pattern example1
Router(config-profile)# pattern example3
```

The following shows an example of a parameter map specifying URL metacharacters to be blocked:

```
Router(config)# parameter-map type urlf-glob metacharacter-param
```

Related CommandsR

Command	Description
class-map type urlfilter	Creates a class map that specifies the traffic to which a URL filtering policy applies.
pattern (parameter-map)	Configures a matching pattern that specifies a list of domains, URL keywords, or URL metacharacters that should be allowed or blocked by local URL filtering.

parser view

To create or change a command-line interface (CLI) view and enter view configuration mode, use the **parser view** command in global configuration mode. To delete a view, use the **no** form of this command.

parser view *view-name*

no parser view *view-name*

Syntax Description

view-name	View name, which can include 1 to 30 alphanumeric characters. The *view-name* argument must not have a number as the first character; otherwise, you will receive the following error message: “Invalid view name.”

Defaults

A CLI view does not exist.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.3(7)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
Cisco IOS XE Release 2.1	This command was integrated into Cisco IOS XE Release 2.1.
12.2(33)SXI	This command was integrated into Cisco IOS Release 12.2(33)SXI.

Usage Guidelines

A CLI view is a set of operational commands and configuration capabilities that restrict user access to the CLI and configuration information; that is, a view allows users to define what commands are accepted and what configuration information is visible.

After you have issued the **parser view** command, you can configure the view via the **secret 5** command and the **commands** command.

To invoke the **parser view** command, the system of the user must be set to root view. The root view can be enabled via the **enable view** command.

Examples

The following example shows how to configure two CLI views, “first” and “second”:

```
Router(config)# parser view first
00:11:40:%PARSER-6-VIEW_CREATED:view 'first' successfully created.
Router(config-view)# secret 5 firstpass
Router(config-view)# command exec include show version
Router(config-view)# command exec include configure terminal
Router(config-view)# command exec include all show ip
Router(config-view)# exit
Router(config)# parser view second
```

```
00:13:42:%PARSER-6-VIEW_CREATED:view 'second' successfully created.
Router(config-view)# secret 5 secondpass
Router(config-view)# command exec include-exclusive show ip interface
Router(config-view)# command exec include logout
Router(config-view)# exit
```

After you have successfully created a view, a system message such as the following will be displayed:

```
%PARSER-6-VIEW_CREATED: view ‘first’ successfully created.
```

After you have successfully deleted a view, a system message such as the following will be displayed:

```
%PARSER-6-VIEW_DELETED: view ‘first’ successfully deleted.
```

Related Commands

Command	Description
commands (view)	Adds commands to a CLI view.
secret 5	Associates a CLI view or a superview with a password.

parser view superview

To create a superview and enter view configuration mode, use the **parser view superview** command in global configuration mode. To delete a superview, use the **no** form of this command.

parser view *superview-name* **superview**

no parser view *superview-name* **superview**

Syntax Description

superview-name	Superview name, which can include 1 to 30 alphanumeric characters. The *superview-name* argument must not have a number as the first character.

Defaults

A superview does not exist.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.3(11)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
Cisco IOS XE Release 2.1	This command was integrated into Cisco IOS XE Release 2.1.
12.2(33)SXI	This command was integrated into Cisco IOS Release 12.2(33)SXI.

Usage Guidelines

A superview consists of one or more command-line interface (CLI) views, which allow users to define what commands are accepted and what configuration information is visible. Superviews allow a network administrator to easily assign all users within configured CLI views to a superview instead of having to assign multiple CLI views to a group of users.

Superviews contain the following characteristics:

- A CLI view can be shared among multiple superviews.
- Commands cannot be configured for a superview; that is, you must add commands to the CLI view and add that CLI view to the superview.
- Users who are logged in to a superview can access all of the commands that are configured for any of the CLI views that are part of the superview.
- Each superview has a password that is used to switch between superviews or from a CLI view to a superview.

Adding CLI Views to a Superview

You can add a view to a superview only after a password has been configured for the superview (via the **secret 5** command). Thereafter, issue the **view** command in view configuration mode to add at least one CLI view to the superview.

Note Before adding a CLI view to a superview, ensure that the CLI views that are added to the superview are valid views in the system; that is, the views have been successfully created via the **parser view** command.

Examples

The following example shows how to create a superview (su_view1) and enter view configuration mode; two CLI views (view_one, view_two) are added to the superview also:

```
Router> enable view
Router# configure terminal
Router(config)# parser view su_view1 superview
Router(config-view)# secret 5 secret
Router(config-view)# view view_one
Router(config-view)# view view_two
```

Related Commands

Command	Description
parser view	Creates or changes a CLI view and enters view configuration mode.
secret 5	Associates a CLI view or a superview with a password.
view	Adds a normal CLI view to a superview.

pass

To allow packets to be sent to the router without being inspected, use the **pass** command in policy-map-class configuration mode.

pass

Syntax Description

This command has no arguments or keywords.

Command Default

Traffic is not passed; that is, it is dropped.

Command Modes

Policy-map-class configuration

Command History

Release	Modification
12.4(6)T	This command was introduced.

Usage Guidelines

You can use this command only after entering the **policy-map type inspect**, **class type inspect**, and **parameter-map type inspect** commands.

Examples

The following example specifies that policy map p1 will pass the traffic:

```
policy-map type inspect p1
 class type inspect c1
  pass
```

Related Commands

Command	Description
class type inspect	Specifies the traffic (class) on which an action is to be performed.
parameter-map type inspect	Configures an inspect parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action.
policy-map type inspect	Creates a Layer 3 or Layer 4 inspect type policy map.

passive

To move a group member directly into passive mode, use the **passive** command in crypto gdoi group configuration mode. To disable the passive mode setting, use the **no** form of this command.

passive

no passive

Syntax Description

This command has no arguments or keywords.

Command Default

The group member is in full crypto send and receive mode.

Command Modes

Crypto gdoi group configuration (crypto-gdoi-group)

Command History

Release	Modification
12.4(22)T	This command was introduced.
Cisco IOS XE Release 2.3	This command was implemented on the Cisco ASR 1000 series routers.

Usage Guidelines

By using the **passive** command, you avoid having to use the **crypto gdoi gm ipsec direction inbound optional** privileged EXEC command, which is not persistent after a router reload and can be overriden by key server configuration from a rekey.

Examples

The following example shows that the group member group1 is being moved to passive mode:

```
crypto gdoi group group1
 identity 2345
 passive
 server address ipv4 10.34.255.57
```

Related Commands

Command	Description
crypto gdoi gm	Changes the IPsec SA status of group members.

password (ca-trustpoint)

To specify the revocation password for the certificate, use the **password** command in ca-trustpoint configuration mode. To erase any stored passwords, use the **no** form of this command.

password *string*

no password

Syntax Description

string	Name of the password.

Defaults

You are prompted for the password during certificate enrollment.

Command Modes

Ca-trustpoint configuration

Command History

Release	Modification
12.2(8)T	This command was introduced.
12.4(24)T	Support for IPv6 Secure Neighbor Discovery (SeND) was added.

Usage Guidelines

Before you can issue the **password** command, you must enable the **crypto ca trustpoint** command, which declares the certification authority (CA) that your router should use and enters ca-trustpoint configuration mode.

This command allows you to specify the revocation password for the certificate before actual certificate enrollment begins. The specified password is encrypted when the updated configuration is written to NVRAM by the router.

If this command is enabled, you will not be prompted for a password during certificate enrollment.

Examples

The following example shows how to specify the password “revokeme” for the certificate request:

```
crypto ca trustpoint trustpoint1
 enrollment url http://trustpoint1.example.com/
 subject-name OU=Spiral Dept., O=example1.com
 ip-address ethernet-0
 auto-enroll regenerate
 password revokeme
```

Related Commands

Command	Description
crypto ca trustpoint	Declares the CA that your router should use.

password (dot1x credentials)

To specify the password for an 802.1X credentials profile, use the **password** command in dot1x credentials configuration mode. To remove the password, use the **no** form of this command.

password [**0** | **7**] *password*

no password

Syntax Description

0	(Optional) A plain text password will follow. The default is 0.
7	(Optional) An encrypted password will follow. The default is 0.
password	The password.

Command Default

A password is not specified.

Command Modes

Dot1x credentials configuration

Command History

Release	Modification
12.4(6)T	This command was introduced.

Usage Guidelines

Before using this command, the **dot1x credentials** command must have been configured.

Examples

The following example shows which credentials profile should be used when configuring a supplicant. The password is “secret.”

```
dot1x credentials basic-user
 username router
 password secret
 description This credentials profile should be used for most configured ports
```

The credentials structure can be applied to an interface along with the **dot1x pae supplicant** command and keyword to enable supplicant functionality on that interface.

```
interface fastethernet 0/1
 dot1x credentials basic-user
 dot1x pae supplicant
```

Related Commands

Command	Description
dot1x credentials	Specifies the 802.1X credentials profile to be used.

password (line configuration)

To specify a password on a line, use the **password** command in line configuration mode. To remove the password, use the **no** form of this command.

password *password*

no password

Syntax Description

password	Character string that specifies the line password. The first character cannot be a number. The string can contain any alphanumeric characters, including spaces, up to 80 characters. You cannot specify the password in the format number-space-anything. The space after the number causes problems. For example, hello 21 is a legal password, but 21 hello is not. The password checking is case sensitive. For example, the password Secret is different than the password secret.

Defaults

No password is specified.

Command Modes

Line configuration

Command History

Release	Modification
10.0	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS release 12.(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

When an EXEC process is started on a line with password protection, the EXEC prompts for the password. If the user enters the correct password, the EXEC prints its normal privileged prompt. The user can try three times to enter a password before the EXEC exits and returns the terminal to the idle state.

Examples

The following example removes the password from virtual terminal lines 1 to 4:

```
line vty 1 4
 no password
```

Related Commands

Command	Description
enable password	Sets a local password to control access to various privilege levels.

password 5

Note Effective with Cisco IOS Release 12.3(14)T, this command is replaced by the **secret** command.

To associate a command-line interface (CLI) view or a superview with a password, use the **password 5** command in view configuration mode.

password 5 *password*

Syntax Description

password	Password for users to enter the CLI view or superview. A password can contain any combination of alphanumeric characters. **Note** The password is case sensitive.

Defaults

A user cannot access a CLI view or superview.

Command Modes

View configuration

Command History

Release	Modification
12.3(7)T	This command was introduced.
12.3(11)T	This command was enhanced to support superviews.
12.3(14)T	This command was replaced by the secret command.

Usage Guidelines

A user cannot access any commands within the CLI view or superview until the **password 5** command has been issued.

Examples

The following example show how to configure two CLI views, “first” and “second” and associate each view with a password:

```
Router(config)# parser view first
00:11:40:%PARSER-6-VIEW_CREATED:view 'first' successfully created.
Router(config-view)# password 5 firstpass
Router(config-view)# command exec include show version
Router(config-view)# command exec include configure terminal
Router(config-view)# command exec include all show ip
Router(config-view)# exit
Router(config)# parser view second
00:13:42:%PARSER-6-VIEW_CREATED:view 'second' successfully created.
Router(config-view)# password 5 secondpass
Router(config-view)# command exec include-exclusive show ip interface
Router(config-view)# command exec include logout
Router(config-view)# exit
```

Related Commands

Command	Description
parser view	Creates or changes a CLI view and enters view configuration mode.

password encryption aes

To enable a type 6 encrypted preshared key, use the **password encryption aes** command in global configuration mode. To disable password encryption, use the **no** form of this command.

password encryption aes

no password encryption aes

Syntax Description

This command has no arguments or keywords.

Defaults

Preshared keys are not encrypted.

Command Modes

Global configuration

Command History

Release	Modification
12.3(2)T	This command was introduced.
12.2(18)SXD	This command was integrated into Cisco IOS Release 12.2(18)SXD.
12.2(33)SRA	This command was integrated into Cisco IOS release 12.(33)SRA.

Usage Guidelines

You can securely store plain text passwords in type 6 format in NVRAM using a command-line interface (CLI). Type 6 passwords are encrypted. Although the encrypted passwords can be seen or retrieved, it is difficult to decrypt them to find out the actual password. Use the **key config-key password-encryption** command with the **password encryption aes** command to configure and enable the password (symmetric cipher Advanced Encryption Standard [AES] is used to encrypt the keys). The password (key) configured using the **key config-key password-encryption** command is the master encryption key that is used to encrypt all other keys in the router.

If you configure the **password encryption aes** command without configuring the **key config-key password-encryption** command, the following message is printed at startup or during any nonvolatile generation (NVGEN) process, such as when the **show running-config** or **copy running-config startup-config** commands have been configured:

```
"Can not encrypt password. Please configure a configuration-key with 'key config-key'"
```

Note For Cisco 836 routers, please note that support for Advanced Encryption Standard (AES) is available only on IP plus images.

Changing a Password

If the password (master key) is changed, or reencrypted, using the **key config-key password-encryption** command), the list registry passes the old key and the new key to the application modules that are using type 6 encryption.

Deleting a Password

If the master key that was configured using the **key config-key password-encryption** command is deleted from the system, a warning is printed (and a confirm prompt is issued) that states that all type 6 passwords will become useless. As a security measure, after the passwords have been encrypted, they will never be decrypted in the Cisco IOS software. However, passwords can be reencrypted as explained in the previous paragraph.

Caution If the password configured using the **key config-key password-encryption** command is lost, it cannot be recovered. The password should be stored in a safe location.

Unconfiguring Password Encryption

If you later unconfigure password encryption using the **no password encryption aes** command, all existing type 6 passwords are left unchanged, and as long as the password (master key) that was configured using the **key config-key password-encryption** command exists, the type 6 passwords will be decrypted as and when required by the application.

Storing Passwords

Because no one can "read" the password (configured using the **key config-key password-encryption** command), there is no way that the password can be retrieved from the router. Existing management stations cannot "know" what it is unless the stations are enhanced to include this key somewhere, in which case the password needs to be stored securely within the management system. If configurations are stored using TFTP, the configurations are not standalone, meaning that they cannot be loaded onto a router. Before or after the configurations are loaded onto a router, the password must be manually added (using the **key config-key password-encryption** command). The password can be manually added to the stored configuration but is not recommended because adding the password manually allows anyone to decrypt all passwords in that configuration.

Configuring New or Unknown Passwords

If you enter or cut and paste cipher text that does not match the master key, or if there is no master key, the cipher text is accepted or saved, but an alert message is printed. The alert message is as follows:

```
"ciphertext>[for username bar>] is incompatible with the configured master key."
```

If a new master key is configured, all the plain keys are encrypted and made type 6 keys. The existing type 6 keys are not encrypted. The existing type 6 keys are left as is.

If the old master key is lost or unknown, you have the option of deleting the master key using the **no key config-key password-encryption** command. Deleting the master key using the **no key config-key password-encryption** command causes the existing encrypted passwords to remain encrypted in the router configuration. The passwords will not be decrypted.

Examples

The following example shows that a type 6 encrypted preshared key has been enabled:

```
Router (config)# password encryption aes
```

RelatedCommands

Command	Description
key config-key password-encryption	Stores a type 6 encryption key in private NVRAM.
password logging	Provides a log of debugging output for a type 6 password operation.

password logging

To get a log of debugging output for a type 6 password operation, use the **password logging** command in global configuration mode. To disable the debugging, use the **no** form of this command.

password logging

no password logging

Syntax Description

This command has no arguments or keywords.

Defaults

Debug logging is not enabled.

Command Modes

Global Configuration #

Command History

Release	Modification
12.3(2)T	This command was introduced.
12.2(18)SXD	This command was integrated into Cisco IOS Release 12.2(18)SXD.
12.2(33)SRA	This command was integrated into Cisco IOS release 12.(33)SRA.

Examples

The following example shows that debug logging is configured:

```
Router# password logging
```

Related Commands

Command	Description
key config-key password-encryption	Stores an encryption key in private NVRAM.
password encryption aes	Enables a type 6 encrypted preshared key.

pattern (parameter-map)

To configure a matching pattern that specifies a list of domains, URL keywords, or URL metacharacters that should be allowed or blocked by local URL filtering, use the **pattern** command in parameter map configuration mode. To delete the parameter map, use the **no** form of this command.

pattern *expression*

no pattern *expression*

Syntax Description

expression	Matching pattern argument that can refer to a domain name, URL keyword, URL metacharacter entry, or URL keyword and URL metacharacter combination.

Command Default

No pattern is created for the parameter map.

Command Modes

Parameter map configuration (config-profile)

Command History

Release	Modification
12.4(15)XZ	This command was introduced.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.

Usage Guidelines

The matching pattern expression is configured for a parameter map created by the **parameter-map type urlf-glob** command. In the pattern expression, the characters /, {, and } are not allowed in the expression. The question mark (?) is not allowed because it is reserved for the help function in the command-line interface (CLI).

URL pattern matching is improved because the period (.) is interpreted as a dot, and not as a wildcard entry representing a single character, as is the case with regex regular expression pattern matching.

A URL keyword is a complete word that occurs after the domain name and that is between the forward slash (/) path delimiters. For example in the URL http://www.example.com/hack/123.html, only “hack” and “123.html” are treated as keywords. Anything in the host or domain name can be allowed or blocked using a domain name, and thus a URL keyword should be a word that comes after the domain name. The entire keyword in the URL must match the pattern. For example if you have **pattern** *hack*, the URL www.example.com/hacksite/123.html doesn't match the pattern. In order to match this URL, you must have *hacksite*.

URL metacharacters allow pattern matching of single characters or ranges of characters to URLs, similar to the way a UNIX style glob expression works. The URL metacharacters are presented in Table 48.

Table 49 URL Metacharacters for URL Pattern Matching

Character	Description
*	Asterisk—matches any sequence of 0 or more characters.
[*abc*]	Character class—matches any character in the brackets. The character matching is case sensitive. For example, [abc] matches a, b, or c.
[*a-c*]	Character range class. Matches any character in the range. The character matching is case sensitive. [a-z] matches any lowercase letter. You can mix characters and ranges; for example, [abcq-z] matches a, b, c, q, r, s, t, u, v, w, x, y, z, and so does [a-cq-z]. **Note** The dash (-) character is literal only if it is the last or the first character within the brackets, [abc-] or [-abc].
[*0-9*]	Numerical range class. Matches any number in the brackets. For example [0-9] matches 0, 1, 2, 3, 4, 5, 6, 7, 8, or 9.

URL metacharacters are combined with domain names and URL keywords for pattern matching. For example, **pattern** *.example.com will match the domain name www.example.com and **pattern** www.[ey]xample.com can be used to block both www.example.com and www.yxample.com. Also, **pattern** www.example[0-9][0-9].com can be used to block www.example01.com, www.example33.com, and www.example99.com. An example of combining a keyword and metacharacter for pattern matching is using **pattern** hack* to block www.example.com/hacksite/123.html.

Examples

The following shows an example of specifying the parameter map for trusted domains:

```
Router(config)# parameter-map type urlf-glob trusted-domain-param
Router(config-profile)# pattern www.example.com
Router(config-profile)# pattern *.example2.com
```

The following shows an example of a parameter map specifying keywords to be blocked:

```
Router(config)# parameter-map type urlf-glob keyword-param
Router(config-profile)# pattern example1
Router(config-profile)# pattern example3
```

The following shows an example of a parameter map specifying URL metacharacters to be blocked:

```
Router(config)# parameter-map type urlf-glob metacharacter-param
Router(config-profile)# pattern www.example[4-9].com
```

Related Commands

Command	Description
class-map type urlfilter	Creates a class map that specifies the traffic to which a URL filtering policy applies.
parameter-map type urlf-glob	Creates or modifies a parameter map used to specify a list of domains, URL keywords, or URL metacharacters that should be allowed or blocked by local URL filtering and enters parameter map configuration mode.

peer address ipv4

To configure a Group Domain of Interpretation (GDOI) redundant peer key server, use the **peer address ipv4** command in GDOI redundancy configuration mode. To remove the peer key server that was configured, use the **no** form of this command.

peer address ipv4 *ip-address*

no peer address ipv4 *ip-address*

Syntax Description

ip-address	IP address of the peer key server.

Command Default

(Redundancy does not function correctly if at least one peer is not configured under the local key server configuration on a key server.)

Command Modes

GDOI redundancy configuration (gdoi-coop-ks-config)

Command History

Release	Modification
12.4(11)T	This command was introduced.
Cisco IOS XE Release 2.3	This command was implemented on the Cisco ASR 1000 series routers.

Usage Guidelines

For redundancy between key servers to operate correctly, there have to be at least two key servers in a redundant group. Therefore, at least one other peer must be defined on a key server using the **peer address ipv4** command. The local key server sets up an Internet Key Exchange (IKE) session with the peer that is defined using this command and proceeds to communicate using IKE informational messages to complete the election process using the specified IP address of the peer.

Examples

The following example shows that two peer key servers have been configured: 10.41.2.5 and 10.33.5.6.

```
address ipv4 10.1.1.1
redundancy
 local priority 10
 peer address ipv4 10.41.2.5
 peer address ipv4 10.33.5.6
```

Related Commands

Command	Description
address ipv4	Sets the source address, which is used as the source for packets originated by the local key server.
local priority	Sets the local key server priority.

Command	Description
redundancy	Enters GDOI redundancy configuration mode and allows for key server redundancy.
server local	Designates a device as a GDOI key server and enters GDOI local server configuration mode.

peer (IKEv2 keyring)

To define a peer or a peer group for the Internet Key Exchange Version 2 (IKEv2) keyring, use the **peer** command in IKEv2 keyring configuration mode. To remove the peer, use the **no** form of this command.

peer *name*

no peer *name*

Syntax Description

name	The peer name.

Command Default

A peer is not defined or configured.

Command Modes

IKEv2 keyring configuration (config-ikev2-keyring)

Command History

Release	Modification
15.1(1)T	This command was introduced.
Cisco IOS XE Release 3.3S	This command was integrated into Cisco IOS XE Release 3.3S.

Usage Guidelines

Use this command to define the name of a peer or peer group. This command enters IKEv2 keyring peer configuration mode. A peer subblock identifies a peer or peer-group using identity, hostname or address statements. A peer subblock must have atleast one statement identifying a peer or peer group. A peer subblock can have a single statement of each type identifying a peer or peer group. A peer subblock can have a single key or key-pair.

Examples

The following example shows how to configure an IKEv2 keyring with multiple peer subblocks:

```
Router(config)# crypto ikev2 keyring keyring-1
Router(configikev2-keyring)# peer peer1
Router(config-ikev2-keyring-peer)# description peer1
Router(config-ikev2-keyring-peer)# address 10.0.0.1
Router(config-ikev2-keyring-peer)# pre-shared-key key-1

Router(configikev2-keyring)# peer peer2
Router(config-ikev2-keyring-peer)# description peer2
Router(config-ikev2-keyring-peer)# host peer1.example.com
Router(config-ikev2-keyring-peer)# pre-shared-key key-2

Router(configikev2-keyring)# peer peer3
Router(config-ikev2-keyring-peer)# description peer3
Router(config-ikev2-keyring-peer)# host peer3.example.com
Router(config-ikev2-keyring-peer)# identity key-id abc
Router(config-ikev2-keyring-peer)# address 10.0.0.3
Router(config-ikev2-keyring-peer)# pre-shared-key key-3
```

Related Commands

Command	Description
address (ikev2 keyring)	Specifies the IPv4 address or the range of the peers in IKEv2 keyring.
crypto ikev2 keyring	Defines an IKEv2 keyring.
description (ikev2 keyring)	Describes an IKEv2 peer or a peer group for the IKEv2 keyring.
hostname (ikev2 keyring)	Specifies the hostname for the peer in the IKEv2 keyring.
identity (ikev2 keyring)	Identifies the peer with IKEv2 types of identity.
pre-shared-key (ikev2 keyring)	Defines a preshared key for the IKEv2 peer.

permit

To set conditions in named IP access list or object group access control list (OGACL) that will permit packets, use the **permit** command in the appropriate configuration mode. To remove a condition from an IP access list or an OGACL, use the **no** form of this command.

permit *protocol* [*source-addr source-wildcard*] {**any** | **host** {*address* | *name*} | **object-group** *object-group-name*} {*destination-addr destination-wildcard* | **any** | **host** {*address* | *name*} | **object-group** *object-group-name*}[**dscp** *dscp-value* | **precendence** *precedence-value*] [**fragments** *fragment-value*] [**option** *option-value*] [**reflect** *access-list-name*] [**time-range** *time-range-value*] [**ttl** *match-value ttl-value* [*ttl-value*]] [**tos** *tos-value*] [**timeout** *max-time*]] [**log** [*log-value*] | **log-input** [*log-input-value*]]

no permit *protocol* [*source-addr source-wildcard*] {**any** | **host** {*address* | *name*} | **object-group** *object-group-name*} {*destination-addr destination-wildcard* | **any** | **host** {*address* | *name*} | **object-group** *object-group-name*}

permit {**tcp** | **udp**} {*source-addr source-wildcard* | **any** | **host** *source-addr* | **object-group** *source-obj-group*} {*destination-addr destination-wildcard* | **any** | **host** *dest-addr* | **object-group** *dest-obj-group* | *port-match-criteria* {*destination-addr destination-wildcard* | **any** | **host** *dest-addr* | **object-group** *dest-obj-group*}} [*port-match-criteria port-number*] [**fragments**] [**ack** | **established**] [**fin**] [**psh**] [**rst**] [**syn**] [**urg**] [**match-all** *match-value* | **match-any** *match-value*] [**dscp** *dscp-value* | **precendence** *precedence-value*] [**option** *option-value*] [**time-range** *time-range-value*] [**ttl** *match-value ttl-value* [*ttl-value*]] [**tos** *tos-value*]] [**log** [*log-value*] | **log-input** [*log-input-value*]]

no permit {**tcp** | **udp**}{*source-addr source-wildcard* | **any** | **host** *source-addr* | **object-group** *source-obj-group*} {*destination-addr destination-wild-card* | **any** | **host** *dest-addr* | **object-group** *dest-obj-group* | *port-match-criteria* {*destination-addr destination-wild-card* | **any** | **host** *dest-addr* | **object-group** *dest-obj-group*}}

Syntax Description

protocol	Name or number of a protocol; valid values are; valid values are **ahp**, **eigrp**, **esp**, **gre**, **icmp**, **igmp**, **igrp**, **ip**, **ipinip**, **nos**, **ospf**, **object-group**, **tcp**, **pcp**, **pim**, **udp**, or an integer in the range 0 to 255 representing an IP protocol number. To match any Internet protocol (including Internet Control Message Protocol (ICMP), TCP, and User Datagram Protocol (UDP), use the keyword **ip**. See the “Usage Guidelines” section for additional qualifiers.
source-addr	(Optional) Number of the network or host from which the packet is being sent in a 32-bit quantity in four-part, dotted-decimal format.
source-wildcard	(Optional) Wildcard bits to be applied to the source in four-part, dotted-decimal format. Place ones in the bit positions you want to ignore.
any	Specifies any source or any destination host as an abbreviation for the *source-addr* or *destination-addr value* and the *source-wildcard* or *destination-wildcard* value of 0.0.0.0 255.255.255.255.
host *address name*	Specifies the source or destination address and name of a single host.
object-group *object-group-name*	Specifies the source or destination name of the object group.
destination-addr	Number of the network or host to which the packet is being sent in a 32-bit quantity in four-part, dotted-decimal format.

destination-wildcard	Wildcard bits to be applied to the destination in a 32-bit quantity in four-part, dotted-decimal format. Place ones in the bit positions you want to ignore.
object-group *dest-addr-group-name*	Specifies the destination address group name.
dscp *dscp-value*	(Optional) Matches the packets with the given Differentiated Services Code Point (DSCP) value; see the "Usage Guidelines" section for valid values.
precedence *precedence-value*	(Optional) Specifies the precedence filtering level for packets; valid values are a number from 0 to 7 or by a name. See the "Usage Guidelines" section for a list of valid names.
fragments *fragment-value*	(Optional) Applies the access list entry to noninitial fragments of packets; the fragment is either permitted or denied accordingly. For more details about the **fragments** keyword, see the "Access List or OGACL Processing of Fragments" and "Fragments and Policy Routing" sections in the "Usage Guidelines" section.
option *option-value*	(Optional) Matches the packets with the given IP options value number; see the "Usage Guidelines" section for valid values.
reflect *access-list-name*	(Optional) Create reflexive access list entry.
time-range *time-range-value*	(Optional) Specifies a time-range entry name.
ttl *match-value ttl-value*	(Optional) Specifies the match packets with given TTL value; see the "Usage Guidelines" section for valid values.
tos *tos-value*	(Optional) Specifies the service filtering level for packets; valid values are a number from 0 to 15 or by a name as listed in the "Usage Guidelines" section of the **access-list** (IP extended) command.
timeout *max-time*	Specifies the maximum time for a reflexive ACL to live; the valid values are from 1 to 2147483 seconds.

log	(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the **logging console** command.) The message for a standard list includes the access list number, whether the packet was permitted or denied, the source address, and the number of packets. The message for an extended list includes the access list number; whether the packet was permitted or denied; the protocol; whether the protocol was TCP, UDP, ICMP, or a number; and, if appropriate, the source and destination addresses and port numbers and the user-defined cookie or router-generated hash value. For both standard and extended lists, the message is generated for the first packet that matches, and then at 5-minute intervals, including the number of packets permitted or denied in the prior 5-minute interval. The logging facility might drop some logging message packets if there are too many to be handled or if there is more than one logging message to be handled in 1 second. This behavior prevents the router from reloading because of too many logging packets. Therefore, the logging facility should not be used as a billing tool or an accurate source of the number of matches to an access list. After you specify the **log** keyword (and the associated *word* argument), you cannot specify any other keywords or settings for this command.
log-value	(Optional) User-defined cookie appended to the log message. The cookie: • cannot be more than characters • cannot start with hexadecimal notation (such as 0x) • cannot be the same as, or a subset of, the following keywords: **reflect**, **fragment**, **time-range** • must contain alphanumeric characters only The user-defined cookie is appended to the access control entry (ACE) syslog entry and uniquely identifies the ACE, within the access control list, that generated the syslog entry.
log-input *log-input-value*	(Optional) Matches the log against this entry, including the input interface. After you specify the **log-input** keyword (and the associated *log-input-value* argument), you cannot specify any other keywords or settings for this command.
tcp	Specifies the TCP protocol.
udp	Specifies the UDP protocol.
object-group *source-obj-group*	Specifies the source address group name.
port-match-criteria port-number	Matches only packets on a given port number; see the “Usage Guidelines” section for valid values.

Command Default

There are no specific conditions under which a packet passes the access list.

Command Modes

Standard access-list configuration (config-std-nacl)
Extended access-list configuration (config-ext-nacl)

Command History

Release	Modification
12.4(20)T	This command was introduced.
12.4(22)T	The *word* argument was added to the **log** and **log-input** keywords.

Usage Guidelines

Use this command following the **ip access-list** command to define the conditions under which a packet passes the access list.

In Cisco IOS 15.0(1)M and later Releases, to remove the log entry from the **permit ip any any log** command, use the **permit ip any any** command.

In releases earlier than Cisco IOS Release15.0(1)M, to remove the **log** option from the **permit ip any any log** command, use the **no permit ip any any log** and the **permit ip any any** commands.

In Cisco IOS 15.0(1)M and later releases, to remove the log entry and the user-defined cookie, use the **permit ip any any** [*log-value*] command.

In releases earlier than Cisco IOS Release 15.0(1)M, to remove the log entry and user-defined cookies, use the **no permit ip any any log** [*log-value*] and **permit ip any any** commands.

Access List or OGACL Processing of Fragments

The behavior of access-list entries regarding the use or lack of the **fragments** keyword are summarized in Table 50:

Table 50 Access list or OGACL Processing of Fragments

If the Access-List Entry Has...	Then...
...no **fragments** keyword (the default behavior), and assuming all of the access-list entry information matches,	For an access-list entry containing only Layer 3 information: • The entry is applied to nonfragmented packets, initial fragments, and noninitial fragments. For an access list entry containing Layer 3 and Layer 4 information: • The entry is applied to nonfragmented packets and initial fragments: – If the entry is a **permit** statement, the packet or fragment is permitted. – If the entry is a **deny** statement, the packet or fragment is denied. • The entry is also applied to noninitial fragments in the following manner. Because noninitial fragments contain only Layer 3 information, only the Layer 3 portion of an access-list entry can be applied. If the Layer 3 portion of the access-list entry matches, and – If the entry is a **permit** statement, the noninitial fragment is permitted. – If the entry is a **deny** statement, the next access-list entry is processed. **Note** The **deny** statements are handled differently for noninitial fragments versus nonfragmented or initial fragments.
...the **fragments** keyword, and assuming all of the access-list entry information matches,	**Note** The access-list entry is applied only to noninitial fragments. The **fragments** keyword cannot be configured for an access-list entry that contains any Layer 4 information.

Ensure that you do not add the **fragments** keyword to every access list entry because the first fragment of the IP packet is considered a nonfragment and is treated independently of the subsequent fragments. An initial fragment will not match an access list **permit** or **deny** entry that contains the **fragments** keyword, the packet is compared to the next access list entry, and so on, until it is either permitted or denied by an access list entry that does not contain the **fragments** keyword. Therefore, you may need two access list entries for every **deny** entry. The first **deny** entry of the pair will not include the **fragments** keyword, and applies to the initial fragment. The second **deny** entry of the pair will include the **fragments** keyword and applies to the subsequent fragments. In the cases where there are multiple

deny access list entries for the same host but with different Layer 4 ports, a single **deny** access-list entry with the **fragments** keyword for that host is all that needs to be added. Thus all the fragments of a packet are handled in the same manner by the access list.

Packet fragments of IP datagrams are considered individual packets and each counts individually as a packet in access list accounting and access list violation counts.

Note The **fragments** keyword cannot solve all cases involving access lists and IP fragments.

Fragments and Policy Routing

Fragmentation and the fragment control feature affect policy routing if the policy routing is based on the **match ip addres**s command and the access list had entries that match on Layer 4 through 7 information. It is possible that noninitial fragments pass the access list and are policy routed, even if the first fragment was not policy routed or the reverse.

By using the **fragments** keyword in access list entries as described earlier, a better match between the action taken for initial and noninitial fragments can be made and it is more likely policy routing will occur as intended.

The *source-addr* and *destination-addr* arguments allow you to create an object group based on a source or destination group. The following keywords and arguments are available:

- **dscp** *dscp-value*—(Optional) Matches the packets with the given DSCP value; the valid values are as follows:
  - **0** to **63**—Differentiated services codepoint value
  - **af11**—Matches the packets with AF11 dscp (001010)
  - **af12**—Matches the packets with AF12 dscp (001100)
  - **af13**—Matches the packets with AF13 dscp (001110)
  - **af21**—Matches the packets with AF21 dscp (010010)
  - **af22**—Matches the packets with AF22 dscp (010100)
  - **af23**—Matches the packets with AF23 dscp (010110)
  - **af31**—Matches the packets with AF31 dscp (011010)
  - **af32**—Matches the packets with AF32 dscp (011100)
  - **af33**—Matches the packets with AF33 dscp (011110)
  - **af41**—Matches the packets with AF41 dscp (100010)
  - **af42**—Matches the packets with AF42 dscp (100100)
  - **af43**—Matches the packets with AF43 dscp (100110)
  - **cs1**—Matches the packets with CS1 (precedence 1) dscp (001000)
  - **cs2**—Matches the packets with CS2 (precedence 2) dscp (010000)
  - **cs3**—Matches the packets with CS3 (precedence 3) dscp (011000)
  - **cs4**—Matches the packets with CS4 (precedence 4) dscp (100000)
  - **cs5**—Matches the packets with CS5 (precedence 5) dscp (101000)
  - **cs6**—Matches the packets with CS6 (precedence 6) dscp (110000)
  - **cs7**—Matches the packets with CS7 (precedence 7) dscp (111000)
  - **default**—Matches the packets with default dscp (000000)

  - **ef**—Matches the packets with EF dscp (101110)
- **fragments**—(Optional) Checks for noninitial fragments. See Table 50.
- **log**—(Optional) Logs the matches against this entry.
- **log-input**—(Optional) Logs the matches against this entry, including the input interface.
- **option** *option-value*—(Optional) Matches the packets with given IP Options value. The valid values are as follows:
  - **0** to **255**—IP Options value.
  - **add-ext**—Matches the packets with Address Extension Option (147).
  - **any-options**—Matches the packets with ANY Option.
  - **com-security**—Matches the packets with Commercial Security Option (134).
  - **dps**—Matches the packets with Dynamic Packet State Option (151).
  - **encode**—Matches the packets with Encode Option (15).
  - **eool**—Matches the packets with End of Options (0).
  - **ext-ip**—Matches the packets with Extended IP Option (145).
  - **ext-security**—Matches the packets with Extended Security Option (133).
  - **finn**—Matches the packets with Experimental Flow Control Option (205).
  - **imitd**—Matches the packets with IMI Traffic Desriptor Option (144).
  - **lsr**—Matches the packets with Loose Source Route Option (131).
  - **match-all**—Matches the packets if all specified flags are present.
  - **match-any**—Matches the packets if any specified flag is present.
  - **mtup**—Matches the packets with MTU Probe Option (11).
  - **mtur**—Matches the packets with MTU Reply Option (12).
  - **no-op**—Matches the packets with No Operation Option (1).
  - **psh**—Match the packets on the PSH bit.
  - **nsapa**—Matches the packets with NSAP Addresses Option (150).
  - **reflect**—Creates reflexive access list entry.
  - **record-route**—Matches the packets with Record Route Option (7).
  - **rst**—Matches the packets on the RST bit.
  - **router-alert**—Matches the packets with Router Alert Option (148).
  - **sdb**—Matches the packets with Selective Directed Broadcast Option (149).
  - **security**—Matches the packets with Basic Security Option (130).
  - **ssr**—Matches the packets with Strict Source Routing Option (137).
  - **stream-id**—Matches the packets with Stream ID Option (136).
  - **syn**—Matches the packets on the SYN bit.
  - **timestamp**—Matches the packets with Time Stamp Option (68).
  - **traceroute**—Matches the packets with Trace Route Option (82).
  - **ump**—Matches the packets with Upstream Multicast Packet Option (152).
  - **visa**—Matches the packets with Experimental Access Control Option (142).

  - **zsu**—Matches the packets with Experimental Measurement Option (10).
- **precedence** *precedence-value*—(Optional) Matches the packets with given precedence value; the valid values are as follows:
  - **0** to **7**—Precedence value.
  - **critical**—Matches the packets with critical precedence (5).
  - **flash**—Matches the packets with flash precedence (3).
  - **flash-override**—Matches the packets with flash override precedence (4).
  - **immediate**—Matches the packets with immediate precedence (2).
  - **internet**—Matches the packets with internetwork control precedence (6).
  - **network**—Matches the packets with network control precedence (7).
  - **priority**—Matches the packets with priority precedence (1).
  - **routine**—Matches the packets with routine precedence (0).
- **reflect acl-name**—(Optional) Creates reflexive access list entry.
- **ttl** *match-value ttl-value*—(Optional) Specifies the match packets with given TTL value; the valid values are as follows:
  - **eq**—Matches packets on a given TTL number.
  - **gt**—Matches packets with a greater TTL number.
  - **lt**—Matches packets with a lower TTL number.
  - **neq**—Matches packets not on a given TTL number.
  - **range**—Matches packets in the range of TTLs.
- **time-range** *time-range-value*—(Optional) Specifies a time-range entry name.
- **tos**—(Optional) Matches the packets with given ToS value; the valid values are as follows:
  - **0** to **15**—Type of service value.
  - **max-reliability**—Matches the packets with the maximum reliable ToS (2).
  - **max-throughput**—Matches the packets with the maximum throughput ToS (4).
  - **min-delay**—Matches the packets with the minimum delay ToS (8).
  - **min-monetary-cost**—Matches the packets with the minimum monetary cost ToS (1).
  - **normal**—Matches the packets with the normal ToS (0).
- **timeout** *max-time*—(Optional) Specifies the maximum time for a reflexive ACL to live; the valid values are from 1 to 2147483 seconds.

Examples

The following example shows how to create an access list that permits packets from the users in my_network_object_group if the protocol ports match the ports specified in my_network_object_group:

```
Router> enable
Router# configure terminal
Router(config)# ip access-list extended my_ogacl_policy
Router(config-ext-nacl)# permit tcp object-group my_network_object_group portgroup
my_service_object_group any
```

The following example shows how to create an access list that permits packets from the users in my_network_object_group if the protocol ports match the ports specified in my_network_object_group. In addition, logging is enabled for the access list, and all syslog entries for this ACE include the word MyServiceCookieValue:

```
Router> enable
Router# configure terminal
Router(config)# ip access-list extended my_ogacl_policy
Router(config-ext-nacl)# permit tcp object-group my_network_object_group portgroup
my_service_object_group any log MyServiceCookieValue
```

Related Commands

Command	Description
deny	Sets conditions in a named IP access list or OGACL that will deny packets.
ip access-group	Applies an ACL or OGACL to an interface or a service policy map.
ip access-list	Defines an IP access list or OGACL by name or number.
ip access-list logging hash-generation	Enables hash value generation for ACE syslog entries.
object-group network	Defines network object groups for use in OGACLs.
object-group service	Defines service object groups for use in OGACLs.
show ip access-list	Displays the contents of IP access lists or OGACLs.
show object-group	Displays information about object groups that are configured.

permit (Catalyst 6500 series switches)

To set conditions for a named IP access list, use the **permit** command in access-list configuration mode. To remove a condition from an access list, use the **no** form of this command.

permit *protocol* {{*source-addr source-wildcard*} | **addrgroup** *object-group-name* | **any** | **host** {*address* | *name*}} {*destination-addr destination-wildcard*} | **addrgroup** *object-group-name* | **any** | **host** {*address* | *name*}}

permit {**tcp** | **udp**} {{*source-addr source-wildcard*} | **addrgroup** *source-addr-group-name* | **any** | **host** {*address* | *name*} {*destination-addr destination-wildcard* | **any** | **eq** *port* | **gt** *port* | **host** {*address* | *name*} | **lt** *port* | **neq** *port* | **portgroup** *srcport-groupname*} {**addrgroup** *dest-addr-groupname* | *destination* | *destination-addr destination-wildcard* | **any** | **eq** *port* | **gt** *port* | **host** {*address* | *name*} | **lt** *port* | **neq** *port* | **portgroup** *destport-groupname*} [**dscp** *type*] [**fragments**] [**option** *option*] [**precedence** *precedence*] [**time-range** *time-range-name*] [**tos** *tos*]] [**log** [*word*] | **log-input** [*word*]]}

no permit *protocol* {{*source-addr source-wildcard*} | **addrgroup** *object-group-name* | **any** | **host** {*address* | *name*}} {*destination-addr destination-wildcard*} | **addrgroup** *object-group-name* | **any** | **host** {*address* | *name*}}

no permit {**tcp** | **udp**} {{*source-addr source-wildcard*} | **addrgroup** *source-addr-group-name* | **any** | **host** {*address* | *name*} {*destination-addr destination-wildcard* | **any** | **eq** *port* | **gt** *port* | **host** {*address* | *name*} | **lt** *port* | **neq** *port* | **portgroup** *srcport-groupname*} {**addrgroup** *dest-addr-groupname* | *destination* | *destination-addr destination-wildcard* | **any** | **eq** *port* | **gt** *port* | **host** {*address* | *name*} | **lt** *port* | **neq** *port* | **portgroup** *destport-groupname*} [**dscp** *type*] [**fragments**] [**option** *option*] [**precedence** *precedence*] [**time-range** *time-range-name*] [**tos** *tos*]] [**log** [*word*] | **log-input** [*word*]]}

Syntax Description

protocol	Name or number of a protocol; valid values are **eigrp**, **gre**, **icmp**, **igmp**, **igrp**, **ip**, **ipinip**, **nos**, **ospf**, **tcp**, or **udp**, or an integer in the range 0 to 255 representing an IP protocol number. To match any Internet protocol (including Internet Control Message Protocol (ICMP), TCP, and User Datagram Protocol (UDP), use the keyword **ip**. See the "Usage Guidelines" section for additional qualifiers.
source-addr	Number of the network or host from which the packet is being sent in a 32-bit quantity in four-part, dotted-decimal format.
source-wildcard	Wildcard bits to be applied to source in four-part, dotted-decimal format. Place ones in the bit positions you want to ignore.
addrgroup *object-group-name*	Specifies the source or destination name of the object group.
any	Specifies any source or any destination host as an abbreviation for the *source-addr* or *destination-addr value* and the *source-wildcard* or *destination-wildcard* value of 0.0.0.0 255.255.255.255.
host *address*	Specifies the source or destination address of a single host.
host *name*	Specifies the source or destination name of a single host.
tcp	Specifies the TCP protocol.
udp	Specifies the UDP protocol.

addrgroup *source-addr-group-name*	Specifies the source address group name.
destination-addr	Number of the network or host to which the packet is being sent in a 32-bit quantity in four-part, dotted-decimal format.
destination-wildcard	Wildcard bits to be applied to the destination in a 32-bit quantity in four-part, dotted-decimal format. Place ones in the bit positions you want to ignore.
eq *port*	Matches only packets on a given port number; see the "Usage Guidelines" section for valid values.
gt *port*	Matches only the packets with a greater port number; see the "Usage Guidelines" section for valid values.
lt *port*	Matches only the packets with a lower port number; see the "Usage Guidelines" section for valid values.
neq *port*	Matches only the packets that are not on a given port number; see the "Usage Guidelines" section for valid values.
portgroup *srcport-group-name*	Specifies the source port object group name.
addrgroup *dest-addr-group-name*	Specifies the destination address group name.
portgroup *destport-group-name*	Specifies the destination port object group name.
dscp *type*	(Optional) Matches the packets with the given Differentiated Services Code Point (DSCP) value; see the "Usage Guidelines" section for valid values.
fragments	(Optional) Applies the access list entry to noninitial fragments of packets; the fragment is either permitted or denied accordingly. For more details about the **fragments** keyword, see the "Access List Processing of Fragments" and "Fragments and Policy Routing" sections in the "Usage Guidelines" section.
option *option*	(Optional) Matches the packets with the given IP options value number; see the "Usage Guidelines" section for valid values.
precedence *precedence*	(Optional) Specifies the precedence filtering level for packets; valid values are a number from 0 to 7 or by a name. See the "Usage Guidelines" section for a list of valid names.
time-range *time-range-name*	(Optional) Specifies a time-range entry name.
tos *tos*	(Optional) Specifies the service filtering level for packets; valid values are a number from 0 to 15 or by a name as listed in the "Usage Guidelines" section of the **access-list** (IP extended) command.
option option	(Optional) Matches packets with the IP options value; see the "Usage Guidelines" section for the valid values.
fragments	(Optional) Applies the access list entry to noninitial fragments of packets; the fragment is either permitted or denied accordingly. For more details about the **fragments** keyword, see the "Access List Processing of Fragments" and "Fragments and Policy Routing" sections in the "Usage Guidelines" section.

log	(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the **logging console** command.) The message for a standard list includes the access list number, whether the packet was permitted or denied, the source address, and the number of packets, and if appropriate, the user-defined cookie or router-generated hash value. The message for an extended list includes the access list number; whether the packet was permitted or denied; the protocol; whether the protocol was TCP, UDP, ICMP, or a number; and, if appropriate, the source and destination addresses and source and destination port numbers, and if appropriate, the user-defined cookie or router-generated hash value. For both standard and extended lists, the message is generated for the first packet that matches, and then at 5-minute intervals, including the number of packets permitted or denied in the prior 5-minute interval. The logging facility might drop some logging message packets if there are too many to be handled or if there is more than one logging message to be handled in 1 second. This behavior prevents the router from reloading due to too many logging packets. Therefore, the logging facility should not be used as a billing tool or an accurate source of the number of matches to an access list. After you specify the **log** keyword (and the associated *word* argument), you cannot specify any other keywords or settings for this command.
word	(Optional) User-defined cookie appended to the log message. The cookie: • cannot be more than characters • cannot start with hexadecimal notation (such as 0x) • cannot be the same as, or a subset of, the following keywords: **reflect**, **fragment**, **time-range** • must contain alphanumeric characters only The user-defined cookie is appended to the access control entry (ACE) syslog entry and uniquely identifies the ACE, within the access control list, that generated the syslog entry.
log-input	(Optional) Matches the log against this entry, including the input interface. After you specify the **log-input** keyword (and the associated *word* argument), you cannot specify any other keywords or settings for this command.

Command Default

There are no specific conditions under which a packet passes the named access list.

Command Modes

Access-list configuration (config-ext-nacl)

Command History

Release	Modification
12.2(33)SXH	This command was introduced.
12.4(22)T	The *word* argument was added to the **log** and **log-input** keywords.

Usage Guidelines

Use this command following the **ip access-list** command to define the conditions under which a packet passes the access list.

The **portgroup** keyword appears only when you configure an extended access list.

Access List Processing of Fragments

The behavior of access-list entries regarding the use or lack of the **fragments** keyword are summarized in Table 51:

Table 51 Access list Processing of Fragments

If the Access-List Entry Has...	Then...
...no **fragments** keyword (the default behavior), and assuming all of the access-list entry information matches,	For an access-list entry containing only Layer 3 information: • The entry is applied to nonfragmented packets, initial fragments, and noninitial fragments. For an access list entry containing Layer 3 and Layer 4 information: • The entry is applied to nonfragmented packets and initial fragments: – If the entry is a **permit** statement, the packet or fragment is permitted. – If the entry is a **deny** statement, the packet or fragment is denied. • The entry is also applied to noninitial fragments in the following manner. Because noninitial fragments contain only Layer 3 information, only the Layer 3 portion of an access-list entry can be applied. If the Layer 3 portion of the access-list entry matches, and – If the entry is a **permit** statement, the noninitial fragment is permitted. – If the entry is a **deny** statement, the next access-list entry is processed. **Note** The **deny** statements are handled differently for noninitial fragments versus nonfragmented or initial fragments.
...the **fragments** keyword, and assuming all of the access-list entry information matches,	**Note** The access-list entry is applied only to noninitial fragments.The **fragments** keyword cannot be configured for an access-list entry that contains any Layer 4 information.

Be aware that you should not simply add the **fragments** keyword to every access list entry because the first fragment of the IP packet is considered a nonfragment and is treated independently of the subsequent fragments. An initial fragment will not match an access list **permit** or **deny** entry that contains the **fragments** keyword, the packet is compared to the next access list entry, and so on, until it is either permitted or denied by an access list entry that does not contain the **fragments** keyword. Therefore, you may need two access list entries for every **deny** entry. The first **deny** entry of the pair will not include the **fragments** keyword, and applies to the initial fragment. The second **deny** entry of the pair will include the **fragments** keyword and applies to the subsequent fragments. In the cases where there are multiple **deny** access list entries for the same host but with different Layer 4 ports, a single **deny** access-list entry with the **fragments** keyword for that host is all that needs to be added. Thus all the fragments of a packet are handled in the same manner by the access list.

Packet fragments of IP datagrams are considered individual packets and each counts individually as a packet in access list accounting and access list violation counts.

Note The **fragments** keyword cannot solve all cases involving access lists and IP fragments.

Fragments and Policy Routing

Fragmentation and the fragment control feature affect policy routing if the policy routing is based on the **match ip addres**s command and the access list had entries that match on Layer 4 through 7 information. It is possible that noninitial fragments pass the access list and are policy routed, even if the first fragment was not policy routed or the reverse.

By using the **fragments** keyword in access list entries as described earlier, a better match between the action taken for initial and noninitial fragments can be made and it is more likely policy routing will occur as intended.

The **portgroup** *srcport-groupname or* **portgroup** *destport-groupname* keywords and arguments allow you to create an object group based on a source or destination group. The following keywords and arguments are available:

- **dscp** *value*—(Optional) Matches the packets with the given DSCP value; the valid values are as follows:
  - **0** to **63**—Differentiated services codepoint value
  - **af11**—Matches the packets with AF11 dscp (001010)
  - **af12**—Matches the packets with AF12 dscp (001100)
  - **af13**—Matches the packets with AF13 dscp (001110)
  - **af21**—Matches the packets with AF21 dscp (010010)
  - **af22**—Matches the packets with AF22 dscp (010100)
  - **af23**—Matches the packets with AF23 dscp (010110)
  - **af31**—Matches the packets with AF31 dscp (011010)
  - **af32**—Matches the packets with AF32 dscp (011100)
  - **af33**—Matches the packets with AF33 dscp (011110)
  - **af41**—Matches the packets with AF41 dscp (100010)
  - **af42**—Matches the packets with AF42 dscp (100100)
  - **af43**—Matches the packets with AF43 dscp (100110)
  - **cs1**—Matches the packets with CS1(precedence 1) dscp (001000)

  - **cs2**—Matches the packets with CS2(precedence 2) dscp (010000)
  - **cs3**—Matches the packets with CS3(precedence 3) dscp (011000)
  - **cs4**—Matches the packets with CS4(precedence 4) dscp (100000)
  - **cs5**—Matches the packets with CS5(precedence 5) dscp (101000)
  - **cs6**—Matches the packets with CS6(precedence 6) dscp (110000)
  - **cs7**—Matches the packets with CS7(precedence 7) dscp (111000)
  - **default**—Matches the packets with default dscp (000000)
  - **ef**—Matches the packets with EF dscp (101110)
- **fragments**—(Optional) Checks for noninitial fragments. See the table "Access List Processing of Fragments."
- **log**—(Optional) Logs the matches against this entry.
- **log-input**—(Optional) Logs the matches against this entry, including the input interface; the valid values are as follows:
- **option** *option*—(Optional) Matches the packets with given IP Options value. The valid values are as follows:
  - **0** to **255**—IP Options value.
  - **add-ext**—Matches the packets with Address Extension Option (147).
  - **any-options**—Matches the packets with ANY Option.
  - **com-security**—Matches the packets with Commercial Security Option (134).
  - **dps**—Matches the packets with Dynamic Packet State Option (151).
  - **encode**—Matches the packets with Encode Option (15).
  - **eool**—Matches the packets with End of Options (0).
  - **ext-ip**—Matches the packets with Extended IP Option (145).
  - **ext-security**—Matches the packets with Extended Security Option (133).
  - **finn**—Matches the packets with Experimental Flow Control Option (205).
  - **imitd**—Matches the packets with IMI Traffic Desriptor Option (144).
  - **lsr**—Matches the packets with Loose Source Route Option (131).
  - **match-all**—Matches the packets if all specified flags are present.
  - **match-any**—Matches the packets if any specified flag is present.
  - **mtup**—Matches the packets with MTU Probe Option (11).
  - **mtur**—Matches the packets with MTU Reply Option (12).
  - **no-op**—Matches the packets with No Operation Option (1).
  - **psh**—Match the packets on the PSH bit.
  - **nsapa**—Matches the packets with NSAP Addresses Option (150).
  - **reflect**—Creates reflexive access list entry.
  - **record-route**—Matches the packets with Record Route Option (7).
  - **rst**—Matches the packets on the RST bit.
  - **router-alert**—Matches the packets with Router Alert Option (148).

  - **sdb**—Matches the packets with Selective Directed Broadcast Option (149).
  - **security**—Matches the packets with Basic Security Option (130).
  - **ssr**—Matches the packets with Strict Source Routing Option (137).
  - **stream-id**—Matches the packets with Stream ID Option (136).
  - **syn**—Matches the packets on the SYN bit.
  - **timestamp**—Matches the packets with Time Stamp Option (68).
  - **traceroute**—Matches the packets with Trace Route Option (82).
  - **ump**—Matches the packets with Upstream Multicast Packet Option (152).
  - **visa**—Matches the packets with Experimental Access Control Option (142).
  - **zsu**—Matches the packets with Experimental Measurement Option (10).
- **precedence** *value*—(Optional) Matches the packets with given precedence value; the valid values are as follows:
  - **0** to **7**—Precedence value.
  - **critical**—Matches the packets with critical precedence (5).
  - **flash**—Matches the packets with flash precedence (3).
  - **flash-override**—Matches the packets with flash override precedence (4).
  - **immediate**—Matches the packets with immediate precedence (2).
  - **internet**—Matches the packets with internetwork control precedence (6).
  - **network**—Matches the packets with network control precedence (7).
  - **priority**—Matches the packets with priority precedence (1).
  - **routine**—Matches the packets with routine precedence (0).
- **reflect acl-name** [**timeout** *time*]—(Optional) Creates reflexive access list entry. The **timeout** *time* keyword and argument specify the maximum time for a reflexive ACL to live; the valid values are from 1 to 2147483 seconds.
- **time-range** *name*—(Optional) Specifies a time-range entry name.
- **tos**—(Optional) Matches the packets with given ToS value; the valid values are as follows:
  - **0** to**15**—Type of service value.
  - **max-reliability**—Matches the packets with the maximum reliable ToS (2).
  - **max-throughput**—Matches the packets with the maximum throughput ToS (4).
  - **min-delay**—Matches the packets with the minimum delay ToS (8).
  - **min-monetary-cost**—Matches the packets with the minimum monetary cost ToS (1).
  - **normal**—Matches the packets with the normal ToS (0).

Examples

The following example shows how to create an access list that permits packets from the users in myAG if the protocol ports match the ports specified in myPG:

```
Router(config)# ip access-list extended my-pbacl-policy
Router(config-ext-nacl)# permit tcp addrgroup myAG portgroup myPG any
```

The following example shows how to create an access list that permits packets from the users in myAG if the protocol ports match the ports specified in myPG. The access list is log enabled, and the cookie value is set to myCookie:

```
Router(config)# ip access-list extended my-pbacl-policy
Router(config-ext-nacl)# permit tcp addrgroup myAG portgroup myPG any log myCookie
```

Related Commands

Command	Description
deny (Catalyst 6500 series switches)	Sets conditions for a named IP access list.
ip access-group	Controls access to an interface.
ip access-list	Defines an IP access list by name.
ip access-list logging hash-generation	Enables hash value generation for ACE syslog entries.
show ip access-lists	Displays the contents of all current IP access lists.

permit (IP)

To set conditions to allow a packet to pass a named IP access list, use the **permit** command in access list configuration mode. To remove a permit condition from an access list, use the **no** form of this command.

[*sequence-number*] **permit** *source* [*source-wildcard*]

[*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**ttl** *operator value*] [**time-range** *time-range-name*] [**fragments**] [**log** [*user-defined-cookie*]]

no *sequence-number*

no permit *source* [*source-wildcard*]

no permit *protocol source source-wildcard destination destination-wildcard* [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**ttl** *operator value*] [**time-range** *time-range-name*] [**fragments**] [**log** [*user-defined-cookie*]]

Internet Control Message Protocol (ICMP)

[*sequence-number*] **permit icmp** *source source-wildcard destination destination-wildcard* [*icmp-type* [*icmp-code*] | *icmp-message*] [**precedence** *precedence*] [**tos** *tos*] [**ttl** *operator value*] [**time-range** *time-range-name*] [**fragments**] [**log** [*user-defined-cookie*]]

Internet Group Management Protocol (IGMP)

[*sequence-number*] **permit igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**tos** *tos*] [**ttl** *operator value*] [**time-range** *time-range-name*] [**fragments**] [**log** [*user-defined-cookie*]]

Transmission Control Protocol (TCP)

[*sequence-number*] **permit tcp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**established** | {**match-any** | **match-all**} {**+** | **-**} *flag-name*] [**precedence** *precedence*] [**tos** *tos*] [**ttl** *operator value*] [**time-range** *time-range-name*] [**fragments**] [**log** [*user-defined-cookie*]]

User Datagram Protocol (UDP)

[*sequence-number*] **permit udp** *source source-wildcard* [*operator* [*port*]] *destination destination-wildcard* [*operator* [*port*]] [**precedence** *precedence*] [**tos** *tos*] [**ttl** *operator value*] [**time-range** *time-range-name*] [**fragments**] [**log** [*user-defined-cookie*]]

Syntax Description

sequence-number	(Optional) Sequence number assigned to the permit statement. The sequence number causes the system to insert the statement in that numbered position in the access list.
source	Number of the network or host from which the packet is being sent. There are three alternative ways to specify the source: • Use a 32-bit quantity in four-part dotted-decimal format. • Use the **any** keyword as an abbreviation for a *source* and *source-wildcard* of 0.0.0.0 255.255.255.255. • Use **host** *source* as an abbreviation for a *source* and *source-wildcard* of *source* 0.0.0.0.
source-wildcard	(Optional) Wildcard bits to be applied to the source. There are three alternative ways to specify the source wildcard: • Use a 32-bit quantity in four-part dotted-decimal format. Place 1s in the bit positions that you want to ignore. • Use the **any** keyword as an abbreviation for a *source* and *source-wildcard* of 0.0.0.0 255.255.255.255. • Use **host** *source* as an abbreviation for a *source* and *source-wildcard* of *source* 0.0.0.0.
protocol	Name or number of an Internet protocol. The *protocol* argument can be one of the keywords **eigrp**, **gre**, **icmp**, **igmp**, **ip**, **ipinip**, **nos**, **ospf**, **tcp**, or **udp**, or an integer in the range from 0 to 255 representing an Internet protocol number. To match any Internet protocol (including ICMP, TCP, and UDP), use the **ip** keyword. **Note** When the **icmp**, **igmp**, **tcp,** and **udp** keywords are entered, they must be followed with the specific command syntax that is shown for the ICMP, IGMP, TCP, and UDP forms of the **permit** command. **Note** To configure a packet filter to allow BGP traffic, use protocol **tcp** and specify the port number as 179 or **bgp**.
destination	Number of the network or host to which the packet is being sent. There are three alternative ways to specify the destination: • Use a 32-bit quantity in four-part dotted-decimal format. • Use the **any** keyword as an abbreviation for the *destination* and *destination-wildcard* of 0.0.0.0 255.255.255.255. • Use **host** *destination* as an abbreviation for a *destination* and *destination-wildcard* of *destination* 0.0.0.0.
destination-wildcard	Wildcard bits to be applied to the destination. There are three alternative ways to specify the destination wildcard: • Use a 32-bit quantity in four-part dotted-decimal format. Place 1s in the bit positions that you want to ignore. • Use the **any** keyword as an abbreviation for a *destination* and *destination-wildcard* of 0.0.0.0 255.255.255.255. • Use **host** *destination* as an abbreviation for a *destination* and *destination-wildcard* of *destination* 0.0.0.0.

option *option-name*	(Optional) Packets can be filtered by IP Options, as specified by a number from 0 to 255, or by the corresponding IP Option name, as listed in Table 52 in the “Usage Guidelines” section.
precedence *precedence*	(Optional) Packets can be filtered by precedence level, as specified by a number from 0 to 7 or by a name.
tos *tos*	(Optional) Packets can be filtered by type of service (ToS) level, as specified by a number from 0 to 15, or by a name as listed in the “Usage Guidelines” section of the **access-list** (IP extended) command.
ttl *operator value*	(Optional) Compares the TTL value in the packet to the TTL value specified in this **permit** statement. • The *operator* can be **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), or **range** (inclusive range). • The *value* can range from 0 to 255. • If the operator is **range**, specify two values separated by a space. • For Release 12.0S, if the operator is **eq** or **neq**, only one TTL value can be specified. • For all other releases, if the operator is **eq** or **neq**, as many as 10 TTL values can be specified, separated by a space.
time-range *time-range-name*	(Optional) Name of the time range that applies to this **permit** statement. The name of the time range and its restrictions are specified by the **time-range** and **absolute** or **periodic** commands, respectively.
fragments	(Optional) The access list entry applies to noninitial fragments of packets; the fragment is either permitted or denied accordingly. For more details about the **fragments** keyword, see the “Access List Processing of Fragments” and “Fragments and Policy Routing” sections in the “Usage Guidelines” section.
log	(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the **logging console** command.) After you specify the **log** keyword (and the associated *word* argument), you cannot specify any other keywords or settings for this command.
user-defined-cookie	(Optional) User-defined cookie appended to the log message. The cookie: • Cannot be more than 64 characters. • Cannot start with hexadecimal notation (such as 0x). • Cannot be the same as, or a subset of, the following keywords: **fragment, reflect**, **time-range**. • Must contain alphanumeric characters only. The user-defined cookie is appended to the Allegro Crypto Engine (ACE) syslog entry and uniquely identifies the ACE, within the access control list, that generated the syslog entry.
icmp	Permits only ICMP packets. When you enter the **icmp** keyword, you must use the specific command syntax shown for the ICMP form of the **permit** command.
icmp-type	(Optional) ICMP packets can be filtered by ICMP message type. The type is a number from 0 to 255.

icmp-code	(Optional) ICMP packets that are filtered by ICMP message type can also be filtered by the ICMP message code. The code is a number from 0 to 255.
icmp-message	(Optional) ICMP packets can be filtered by an ICMP message type name or an ICMP message type and code name. The possible names are listed in the "Usage Guidelines" section of the **access-list** (IP extended) command.
igmp	Permits only IGMP packets. When you enter the **igmp** keyword, you must use the specific command syntax shown for the IGMP form of the **permit** command.
igmp-type	(Optional) IGMP packets can be filtered by IGMP message type or message name. A message type is a number from 0 to 15. IGMP message names are listed in the "Usage Guidelines" section of the **access-list** (IP extended) command.
tcp	Permits only TCP packets. When you enter the **tcp** keyword, you must use the specific command syntax shown for the TCP form of the **permit** command.
operator	(Optional) Compares source or destination ports. Operators are **eq** (equal) , **gt** (greater than),**lt** (less than), **neq** (not equal), and **range** (inclusive range). If the operator is positioned after the *source* and *source-wildcard* arguments, it must match the source port. If the operator is positioned after the *destination* and *destination-wildcard* arguments, it must match the destination port. The **range** operator requires two port numbers. Up to ten port numbers can be entered for the **eq** (equal) and **neq** (not equal) operators. All other operators require one port number.
port	(Optional) The decimal number or name of a TCP or UDP port. A port number is a number from 0 to 65535. TCP and UDP port names are listed in the "Usage Guidelines" section of the **access-list** (IP extended) command. TCP port names can be used only when filtering TCP. UDP port names can be used only when filtering UDP.
established	(Optional) For the TCP protocol only: Indicates an established connection. A match occurs if the TCP datagram has the ACK or RST bit set. The nonmatching case is that of the initial TCP datagram to form a connection.
{**match-any** | **match-all**}	(Optional) For the TCP protocol only: A match occurs if the TCP datagram has certain TCP flags set or not set. You use the **match-any** keyword to allow a match to occur if any of the specified TCP flags are present, or you can use the **match-all** keyword to allow a match to occur only if all of the specified TCP flags are present. You must follow the **match-any** and **match-all** keywords with the **+** or **-** keyword and the *flag-name* argument to match on one or more TCP flags.

{**+** | **-**} *flag-name*	(Optional) For the TCP protocol only: The **+** keyword matches IP packets if their TCP headers contain the TCP flags that are specified by the *flag-name* argument. The **-** keyword matches IP packets that do not contain the TCP flags specified by the *flag-name* argument. You must follow the **+** and **-** keywords with the *flag-name* argument. TCP flag names can be used only when filtering TCP. Flag names for the TCP flags are as follows: **ack**, **fin**, **psh**, **rst**, **syn**, and **urg**.
udp	Permits only UDP packets. When you enter the **udp** keyword, you must use the specific command syntax shown for the UDP form of the **permit** command.

Command Default

There are no specific conditions under which a packet passes the named access list.

Command Modes

Access list configuration (config-ext-nacl)

Command History

Release	Modification
11.2	This command was introduced.
12.0(1)T	The **time-range** *time-range-name* keyword and argument were added.
12.0(11)	The **fragments** keyword was added.
12.2(13)T	The **igrp** keyword was removed because the IGRP protocol was no longer available in Cisco IOS software.
12.2(14)S	The *sequence-number* argument was added.
12.2(15)T	The *sequence-number* argument was added.
12.3(4)T	The **option** *option-name* keyword and argument were added. The **match-any**, **match-all, +**, and **-** keywords and the *flag-name* argument were added.
12.3(7)T	Command functionality was modified to allow up to ten port numbers to be added after the **eq** and **neq** operators so that an access list entry can be created with noncontiguous ports.
12.4	The **drip** keyword was added to specify the TCP port number used for Optimized Edge Routing (OER) communication.
12.4(2)T	The **ttl** *operator value* keyword and arguments were added.
12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.4(22)T	The *word* argument was added to the **log** keyword.
Cisco IOS XE Release 3.2	This command was implemented on Cisco ASR 1000 Series Aggregation Services Routers.

Usage Guidelines

Use this command following the **ip access-list** command to define the conditions under which a packet passes the named access list.

The **time-range** keyword allows you to identify a time range by name. The **time-range**, **absolute**, and **periodic** commands specify when this **permit** statement is in effect.

log Keyword

A log message includes the access list number or access list name, and whether the packet was permitted or denied; the protocol, whether it was TCP, UDP, ICMP, or a number; and, if appropriate, the source and destination addresses and port numbers, and the user-defined cookie or router-generated hash value. The message is generated for the first packet that matches, and then at 5-minute intervals, including the number of packets permitted or denied in the prior 5-minute interval.

Use the **ip access-list log-update** command to generate logging messages when the number of matches reaches a configurable threshold (rather than waiting for a 5-minute-interval). See the **ip access-list log-update** command for more information.

The logging facility might drop some logging message packets if there are too many to be handled or if there is more than one logging message to be handled in 1 second. This behavior prevents the router from reloading because of too many logging packets. Therefore, the logging facility should not be used as a billing tool or an accurate source of the number of matches to an access list.

If you enable Cisco Express Forwarding and then create an access list that uses the **log** keyword, the packets that match the access list are not Cisco Express Forwarding switched. They are fast-switched. Logging disables Cisco Express Forwarding .

Access List Filtering of IP Options

Access control lists can be used to filter packets with IP Options to prevent routers from being saturated with spurious packets containing IP Options. To see a complete table of all IP Options, including ones currently not in use, refer to the latest Internet Assigned Numbers Authority (IANA) information that is available from its URL: www.iana.org.

Cisco IOS software allows you to filter packets according to whether they contain one or more of the legitimate IP Options by entering either the IP Option value or the corresponding name for the *option-name* argument as shown in Table 52.

Table 52 IP Option Values and Names

IP Option Value or Name	Description
0 to 255	IP Options values.
add-ext	Match packets with Address Extension Option (147).
any-options	Match packets with any IP Option.
com-security	Match packets with Commercial Security Option (134).
dps	Match packets with Dynamic Packet State Option (151).
encode	Match packets with Encode Option (15).
eool	Match packets with End of Options (0).
ext-ip	Match packets with Extended IP Options (145).
ext-security	Match packets with Extended Security Option (133).
finn	Match packets with Experimental Flow Control Option (205).
imitd	Match packets with IMI Traffic Descriptor Option (144).

Table 52 IP Option Values and Names (continued)

IP Option Value or Name	Description
lsr	Match packets with Loose Source Route Option (131).
mtup	Match packets with MTU Probe Option (11).
mtur	Match packets with MTU Reply Option (12).
no-op	Match packets with No Operation Option (1).
nsapa	Match packets with NSAP Addresses Option (150).
psh	Match the packets on the PSH bit.
record-route	Match packets with Router Record Route Option (7).
reflect	Create reflexive access list entry.
router-alert	Match packets with Router Alert Option (148).
rst	Matche the packets on the RST bit.
sdb	Match packets with Selective Directed Broadcast Option (149).
security	Match packets with Base Security Option (130).
ssr	Match packets with Strict Source Routing Option (137).
stream-id	Match packets with Stream ID Option (136).
syn	Matches the packets on the SYN bit.
timestamp	Match packets with Time Stamp Option (68).
traceroute	Match packets with Trace Route Option (82).
ump	Match packets with Upstream Multicast Packet Option (152).
visa	Match packets with Experimental Access Control Option (142).
zsu	Match packets with Experimental Measurement Option (10).

Filtering IP Packets Based on TCP Flags

The access list entries that make up an access list can be configured to detect and drop unauthorized TCP packets by allowing only the packets that have very specific groups of TCP flags set or not set. Users can select any desired combination of TCP flags with which to filter TCP packets. Users can configure access list entries in order to allow matching on a flag that is set and on a flag that is not set. Use the **+** and **-** keywords with a flag name to specify that a match is made based on whether a TCP header flag has been set. Use the **match-any** and **match-all** keywords to allow the packet if any or all, respectively, of the flags specified by the **+** or **-** keyword and *flag-name* argument have been set or not set.

Permitting Optimized Edge Routing (OER) Communication

The **drip** keyword was introduced under the **tcp** keyword to support packet filtering in a network where OER is configured. The **drip** keyword specifies port 3949 that OER uses for internal communication. This option allows you to build a packet filter that permits communication between an OER master controller and border routers. The **drip** keyword is entered following the TCP source, destination addresses, and the **eq** operator. See the example in the “Examples” section.

Access List Processing of Fragments

The behavior of access list entries regarding the use or lack of use of the **fragments** keyword can be summarized as follows:

If the Access-List Entry Has ...	Then ...
... no **fragments** keyword (the default behavior), and assuming all of the access list entry information matches,	For an access list entry that contains only Layer 3 information, the entry is applied to nonfragmented packets, initial fragments, and noninitial fragments. For an access list entry that contains Layer 3 and Layer 4 information: • The entry is applied to nonfragmented packets and initial fragments. – If the entry is a **permit** statement, then the packet or fragment is permitted. – If the entry is a **deny** statement, then the packet or fragment is denied. • The entry is also applied to noninitial fragments in the following manner. Because noninitial fragments contain only Layer 3 information, only the Layer 3 portion of an access list entry can be applied. If the Layer 3 portion of the access list entry matches, and – If the entry is a **permit** statement, then the noninitial fragment is permitted. – If the entry is a **deny** statement, then the next access list entry is processed. **Note** The **deny** statements are handled differently for noninitial fragments versus nonfragmented or initial fragments.
... the **fragments** keyword, and assuming all of the access list entry information matches,	The access list entry is applied only to noninitial fragments. The **fragments** keyword cannot be configured for an access list entry that contains any Layer 4 information.

Be aware that you should not add the **fragments** keyword to every access list entry because the first fragment of the IP packet is considered a nonfragment and is treated independently of the subsequent fragments. An initial fragment will not match an access list **permit** or **deny** entry that contains the **fragments** keyword. The packet is compared to the next access list entry, and so on, until it is either permitted or denied by an access list entry that does not contain the **fragments** keyword. Therefore, you may need two access list entries for every **deny** entry. The first **deny** entry of the pair will not include the **fragments** keyword and applies to the initial fragment. The second **deny** entry of the pair will include the **fragments** keyword and applies to the subsequent fragments. In the cases in which there are multiple **deny** access list entries for the same host but with different Layer 4 ports, a single **deny** access list entry with the **fragments** keyword for that host is all that needs to be added. Thus all the fragments of a packet are handled in the same manner by the access list.

Packet fragments of IP datagrams are considered individual packets, and each counts individually as a packet in access list accounting and access list violation counts.

Note The **fragments** keyword cannot solve all cases that involve access lists and IP fragments.

Fragments and Policy Routing

Fragmentation and the fragment control feature affect policy routing if the policy routing is based on the **match ip address** command and the access list has entries that match on Layer 4 through 7 information. It is possible that noninitial fragments pass the access list and are policy-routed, even if the first fragment is not policy-routed.

If you specify the **fragments** keyword in access list entries, a better match between the action taken for initial and noninitial fragments can be made, and it is more likely that policy routing will occur as intended.

Creating an Access List Entry with Noncontiguous Ports

For Cisco IOS Release 12.3(7)T and later releases, you can specify noncontiguous ports on the same access control entry, which greatly reduces the number of access list entries required for the same source address, destination address, and protocol. If you maintain large numbers of access list entries, we recommend that you consolidate them when possible by using noncontiguous ports. You can specify up to ten port numbers following the **eq** and **neq** operators.

Examples

The following example shows how to set conditions for a standard access list named Internetfilter:

```
ip access-list standard Internetfilter
 deny 192.168.34.0  0.0.0.255
 permit 172.16.0.0  0.0.255.255
 permit 10.0.0.0  0.255.255.255
! (Note: all other access implicitly denied).
```

The following example shows how to permit Telnet traffic on Mondays, Tuesdays, and Fridays from 9:00 a.m. to 5:00 p.m.:

```
time-range testing
 periodic Monday Tuesday Friday 9:00 to 17:00
!
ip access-list extended legal
 permit tcp any any eq telnet time-range testing
!
interface ethernet0
 ip access-group legal in
```

The following example shows how to set a permit condition for an extended access list named filter2. The access list entry specifies that a packet may pass the named access list only if it contains the NSAP Addresses IP Option, which is represented by the IP Option value nsapa.

```
ip access-list extended filter2
 permit ip any any option nsapa
```

The following example shows how to set a permit condition for an extended access list named kmdfilter1. The access list entry specifies that a packet can pass the named access list only if the RST IP flag has been set for that packet:

```
ip access-list extended kmdfilter1
 permit tcp any any match-any +rst
```

The following example shows how to set a permit condition for an extended access list named kmdfilter1. The access list entry specifies that a packet can pass the named access list if the RST TCP flag or the FIN TCP flag has been set for that packet:

```
ip access-list extended kmdfilter1
 permit tcp any any match-any +rst +fin
```

The following example shows how to verify the access list by using the **show access-lists** command and then to add an entry to an existing access list:

```
Router# show access-lists

Standard IP access list 1
2 permit 10.0.0.0, wildcard bits 0.0.255.255
5 permit 10.0.0.0, wildcard bits 0.0.255.255
10 permit 10.0.0.0, wildcard bits 0.0.255.255
20 permit 10.0.0.0, wildcard bits 0.0.255.255

ip access-list standard 1
 15 permit 10.0.0.0 0.0.255.255
```

The following examples shows how to remove the entry with the sequence number of 20 from the access list:

```
ip access-list standard 1
 no 20

!Verify that the list has been removed.

Router# show access-lists

Standard IP access list 1
10 permit 0.0.0.0, wildcard bits 0.0.0.255
30 permit 0.0.0.0, wildcard bits 0.0.0.255
40 permit 0.4.0.0, wildcard bits 0.0.0.255
```

The following example shows how, if a user tries to enter an entry that is a duplicate of an entry already on the list, no changes occur. The entry that the user is trying to add is a duplicate of the entry already in the access list with a sequence number of 20.

```
Router# show access-lists 101

Extended IP access list 101
    10 permit ip host 10.0.0.0 host 10.5.5.34
    20 permit icmp any any
    30 permit ip host 10.0.0.0 host 10.2.54.2
    40 permit ip host 10.0.0.0 host 10.3.32.3 log

ip access-list extended 101
```

```
 100 permit icmp any any

Router# show access-lists 101

Extended IP access list 101
    10 permit ip host 10.3.3.3 host 10.5.5.34
    20 permit icmp any any
    30 permit ip host 10.34.2.2 host 10.2.54.2
    40 permit ip host 10.3.4.31 host 10.3.32.3 log
```

The following example shows what occurs if a user tries to enter a new entry with a sequence number of 20 when an entry with a sequence number of 20 is already in the list. An error message appears, and no change is made to the access list.

```
Router# show access-lists 101

Extended IP access lists 101
    10 permit ip host 10.3.3.3 host 10.5.5.34
    20 permit icmp any any
    30 permit ip host 10.34.2.2 host 10.2.54.2
    40 permit ip host 10.3.4.31 host 10.3.32.3 log

ip access-lists extended 101
 20 permit udp host 10.1.1.1 host 10.2.2.2

%Duplicate sequence number.

Router# show access-lists 101

Extended IP access lists 101
    10 permit ip host 10.3.3.3 host 10.5.5.34
    20 permit icmp any any
    30 permit ip host 10.34.2.2 host 10.2.54.2
    40 permit ip host 10.3.4.31 host 10.3.32.3 log
```

The following example shows several **permit** statements that can be consolidated into one access list entry with noncontiguous ports. The **show access-lists** command is entered to display a group of access list entries for the access list named aaa.

```
Router# show access-lists aaa

Extended IP access lists aaa
 10 permit tcp any eq telnet any eq 450
 20 permit tcp any eq telnet any eq 679
 30 permit tcp any eq ftp any eq 450
 40 permit tcp any eq ftp any eq 679
```

Because the entries are all for the same **permit** statement and simply show different ports, they can be consolidated into one new access list entry. The following example shows the removal of the redundant access list entries and the creation of a new access list entry that consolidates the previously displayed group of access list entries:

```
ip access-list extended aaa
 no 10
 no 20
 no 30
 no 40
 permit tcp any eq telnet ftp any eq 450 679
```

The following example shows the creation of the consolidated access list entry:

```
Router# show access-lists aaa
```

```
Extended IP access list aaa
 10 permit tcp any eq telnet ftp any eq 450 679
```

The following access list filters IP packets containing Type of Service (ToS) level 3 with TTL values 10 and 20. It also filters IP packets with a TTL greater than 154 and applies that rule to noninitial fragments. It permits IP packets with a precedence level of flash and a TTL not equal to 1, and sends log messages about such packets to the console. All other packets are denied.

```
ip access-list extended canton
 deny ip any any tos 3 ttl eq 10 20
 deny ip any any ttl gt 154 fragments
 permit ip any any precedence flash ttl neq 1 log
```

The following example shows how to configure a packet filter, for any TCP source and destination, that permits communication between an OER master controller and border router:

```
ip access-list extended 100
 permit any any tcp eq drip
 exit
```

The following example shows how to set a permit condition for an extended access list named filter_logging. The access list entry specifies that a packet may pass the named access list only if it is of TCP protocol type and destined to host 10.5.5.5, all other packets are denied. In addition, the logging mechanism is enabled and one of the user defined cookies (Permit_tcp_to_10.5.5.5 or Deny_all) is appended to the appropriate syslog entry.

```
ip access-list extended filter_logging
 permit tcp any host 10.5.5.5 log Permit_tcp_to_10.5.5.5
 deny ip any any log Deny_all
```

The following example shows how to configure a packet filter for any TCP source and destination that permits inbound and outbound BGP traffic:

```
ip access-list extended 100
 permit tcp any eq bgp any eq bgp
```

Related Commands

Command	Description
absolute	Specifies an absolute time when a time range is in effect.
access-list (IP extended)	Defines an extended IP access list.
access-list (IP standard)	Defines a standard IP access list.
deny (IP)	Sets conditions under which a packet does not pass a named IP access list.
ip access-group	Controls access to an interface.
ip access-list log-update	Sets the threshold number of packets that cause a logging message.
ip access-list logging hash-generation	Enables hash value generation for ACE syslog entries.
ip access-list resequence	Applies sequence numbers to the access list entries in an access list.
ip options	Drops or ignores IP Options packets that are sent to the router.
logging console	Sends system logging (syslog) messages to all available TTY lines and limits messages based on severity.

Command	Description
match ip address	Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, or performs policy routing on packets.
periodic	Specifies a recurring (weekly) time range for functions that support the time-range feature.
show access-lists	Displays a group of access-list entries.
show ip access-list	Displays the contents of all current IP access lists.
time-range	Specifies when an access list or other feature is in effect.

permit (MAC ACL)

To set conditions for a MAC access list, use the **permit** command in MAC access-list extended configuration mode. To remove a condition from an access list, use the **no** form of this command.

permit {*src_mac_mask* | {**host name** *src_mac_name*} | **any**} {*dest_mac_mask* | {**host name** *dst_mac_name*} | **any**} [{*protocol_keyword* | {*ethertype_number ethertype_mask*}} [**vlan** *vlan_ID*] [**cos** *cos_value*]]

no permit {*src_mac_mask* | {**host name** *src_mac_name*} | **any**} {*dest_mac_mask* | {**host name** *dst_mac_name*} | **any**} [{*protocol_keyword* | {*ethertype_number ethertype_mask*}} [**vlan** *vlan_ID*] [**cos** *cos_value*]]

Syntax Description

src_mac_mask	Specifies the MAC address mask that identifies a selected block of source MAC addresses. A value of 1 represents a wildcard in that position.
host name *src_mac_name*	Specifies a source host that has been named using the **mac host name** command.
any	Specifies any source or any destination host as an abbreviation for the *src_mac_mask* or *dest_mac_mask* value of 1111.1111.1111, which declares all digits to be wildcards.
dest_mac_mask	Specifies the MAC address mask that identifies a selected block of destination MAC addresses.
host name *dst_mac_name*	Specifies a destination host that has been named using the **mac host name** command.
protocol_keyword	(Optional) Specifies a named protocol (for example, ARP).
ethertype_number	(Optional) The EtherType number specifies the protocol within the Ethernet packet.
ethertype_mask	(Optional) The EtherType mask allows a range of EtherTypes to be specified together. This is a hexadecimal number from 0 to FFFF. An EtherType mask of 0 requires an exact match of the EtherType.
vlan *vlan_ID*	(Optional) Specifies a VLAN.
cos *cos_value*	(Optional) Specifies the Layer 2 priority level for packets. The range is from 0 to 7.

Command Default

This command has no defaults.

Command Modes

MAC access-list extended configuration (config-ext-macl)

Command History

Release	Modification
12.2(33)SXI	This command was introduced.

Usage Guidelines

Use this command following the **ip access-list** command to define the conditions under which a packet passes the access list.

- The **vlan** and **cos** keywords are not supported in MAC ACLs used for VACL filtering.
- The **vlan** keyword for VLAN-based QoS filtering in MAC ACLs can be globally enabled or disabled and is disabled by default.
- Enter MAC addresses as three 2-byte values in dotted hexadecimal format. For example, 0123.4567.89ab.
- Enter MAC address masks as three 2-byte values in dotted hexadecimal format. Use 1 bits as wildcards. For example, to match an address exactly, use 0000.0000.0000 (can be entered as 0.0.0).
- An entry without a protocol parameter matches any protocol.
- Enter an EtherType and an EtherType mask as hexadecimal values from 0 to FFFF.
- This list shows the EtherType values and their corresponding protocol keywords:
  - 0x0600—xns-idp—Xerox XNS IDP
  - 0x0BAD—vines-ip—Banyan VINES IP
  - 0x0baf—vines-echo—Banyan VINES Echo
  - 0x6000—etype-6000—DEC unassigned, experimental
  - 0x6001—mop-dump—DEC Maintenance Operation Protocol (MOP) Dump/Load Assistance
  - 0x6002—mop-console—DEC MOP Remote Console
  - 0x6003—decnet-iv—DEC DECnet Phase IV Route
  - 0x6004—lat—DEC Local Area Transport (LAT)
  - 0x6005—diagnostic—DEC DECnet Diagnostics
  - 0x6007—lavc-sca—DEC Local-Area VAX Cluster (LAVC), SCA
  - 0x6008—amber—DEC AMBER
  - 0x6009—mumps—DEC MUMPS
  - 0x0800—ip—Malformed, invalid, or deliberately corrupt IP frames
  - 0x8038—dec-spanning—DEC LANBridge Management
  - 0x8039—dsm—DEC DSM/DDP
  - 0x8040—netbios—DEC PATHWORKS DECnet NETBIOS Emulation
  - 0x8041—msdos—DEC Local Area System Transport
  - 0x8042—etype-8042—DEC unassigned
  - 0x809B—appletalk—Kinetics EtherTalk (AppleTalk over Ethernet)
  - 0x80F3—aarp—Kinetics AppleTalk Address Resolution Protocol (AARP)

Examples

This example shows how to create a MAC-Layer ACL named mac_layer that permits dec-phase-iv traffic with source address 0000.4700.0001 and destination address 0000.4700.0009, but denies all other traffic:

```
Router(config)# mac access-list extended mac_layer
Router(config-ext-macl)# permit 0000.4700.0001 0.0.0 0000.4700.0009 0.0.0 dec-phase-iv
Router(config-ext-macl)# deny any any
```

Related Commands

Command	Description
deny (MAC ACL)	Sets deny conditions for a named MAC access list.
mac access-list extended	Defines a MAC access list by name.
mac host	Assigns a name to a MAC address.
show mac access-group	Displays the contents of all current MAC access groups.

permit (reflexive)

To create a reflexive access list and to enable its temporary entries to be automatically generated, use the **permit** command in access-list configuration mode. To delete the reflexive access list (if only one protocol was defined) or to delete protocol entries from the reflexive access list (if multiple protocols are defined), use the **no** form of this command.

permit *protocol source source-wildcard destination destination-wildcard* **reflect** *name* [**timeout** *seconds*]

no permit *protocol source-wildcard destination destination-wildcard* **reflect** *name*

Syntax Description

protocol	Name or number of an IP protocol. It can be one of the keywords **gre**, **icmp**, **ip**, **ipinip**, **nos**, **tcp**, or **udp**, or an integer in the range 0 to 255 representing an IP protocol number. To match any Internet protocol (including Internet Control Message Protocol, Transmission Control Protocol, and User Datagram Protocol), use the keyword **ip**.
source	Number of the network or host from which the packet is being sent. There are three other ways to specify the source: • Use a 32-bit quantity in four-part, dotted-decimal format. • Use the keyword **any** as an abbreviation for a *source* and *source-wildcard* of 0.0.0.0 255.255.255.255. This keyword is normally not recommended (see the section “Usage Guidelines”). • Use **host** *source* as an abbreviation for a *source* and *source-wildcard* of *source* 0.0.0.0.
source-wildcard	Wildcard bits (mask) to be applied to source. There are three other ways to specify the source wildcard: • Use a 32-bit quantity in four-part, dotted-decimal format. Place ones in the bit positions you want to ignore. • Use the keyword **any** as an abbreviation for a *source* and *source-wildcard* of 0.0.0.0 255.255.255.255. This keyword is normally not recommended (see the section “Usage Guidelines”). • Use **host** *source* as an abbreviation for a *source* and *source-wildcard* of *source* 0.0.0.0.
destination	Number of the network or host to which the packet is being sent. There are three other ways to specify the destination: • Use a 32-bit quantity in four-part, dotted-decimal format. • Use the keyword **any** as an abbreviation for the *destination* and *destination-wildcard* of 0.0.0.0 255.255.255.255. This keyword is normally *not* recommended (see the section “Usage Guidelines”). • Use **host** *destination* as an abbreviation for a *destination* and *destination-wildcard* of destination 0.0.0.0.

destination-wildcard	Wildcard bits to be applied to the destination. There are three other ways to specify the destination wildcard: • Use a 32-bit quantity in four-part, dotted-decimal format. Place ones in the bit positions you want to ignore. • Use the keyword **any** as an abbreviation for a *destination* and *destination-wildcard* of 0.0.0.0 255.255.255.255. This keyword is normally *not* recommended (see the section "Usage Guidelines"). • Use **host** *destination* as an abbreviation for a *destination* and *destination-wildcard* of *destination* 0.0.0.0.
reflect	Identifies this access list as a reflexive access list.
name	Specifies the name of the reflexive access list. Names cannot contain a space or quotation mark, and must begin with an alphabetic character to prevent ambiguity with numbered access lists. The name can be up to 64 characters long.
timeout *seconds*	(Optional) Specifies the number of seconds to wait (when no session traffic is being detected) before entries expire in this reflexive access list. Use a positive integer from 0 to 2^{32}–1. If not specified, the number of seconds defaults to the global timeout value.

Defaults

If this command is not configured, no reflexive access lists will exist, and no session filtering will occur.

If this command is configured without specifying a **timeout** value, entries in this reflexive access list will expire after the global timeout period.

Command Modes

Access-list configuration

Command History

Release	Modification
11.3	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS release 12.(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

This command is used to achieve reflexive filtering, a form of session filtering.

For this command to work, you must also nest the reflexive access list using the **evaluate** command.

This command creates a reflexive access list and triggers the creation of entries in the same reflexive access list. This command must be an entry (condition statement) in an extended named IP access list.

If you are configuring reflexive access lists for an external interface, the extended named IP access list should be one which is applied to outbound traffic.

If you are configuring reflexive access lists for an internal interface, the extended named IP access list should be one which is applied to inbound traffic.

IP sessions that originate from within your network are initiated with a packet exiting your network. When such a packet is evaluated against the statements in the extended named IP access list, the packet is also evaluated against this reflexive **permit** entry.

As with all access list entries, the order of entries is important, because they are evaluated in sequential order. When an IP packet reaches the interface, it will be evaluated sequentially by each entry in the access list until a match occurs.

If the packet matches an entry prior to the reflexive **permit** entry, the packet will not be evaluated by the reflexive **permit** entry, and no temporary entry will be created for the reflexive access list (session filtering will not be triggered).

The packet will be evaluated by the reflexive **permit** entry if no other match occurs first. Then, if the packet matches the protocol specified in the reflexive **permit** entry, the packet is forwarded and a corresponding temporary entry is created in the reflexive access list (unless the corresponding entry already exists, indicating the packet belongs to a session in progress). The temporary entry specifies criteria that permits traffic into your network only for the same session.

Characteristics of Reflexive Access List Entries

This command enables the creation of temporary entries in the same reflexive access list that was defined by this command. The temporary entries are created when a packet exiting your network matches the protocol specified in this command. (The packet “triggers” the creation of a temporary entry.) These entries have the following characteristics:

- The entry is a **permit** entry.
- The entry specifies the same IP upper-layer protocol as the original triggering packet.
- The entry specifies the same source and destination addresses as the original triggering packet, except the addresses are swapped.
- If the original triggering packet is TCP or UDP, the entry specifies the same source and destination port numbers as the original packet, except the port numbers are swapped.

  If the original triggering packet is a protocol other than TCP or UDP, port numbers do not apply, and other criteria are specified. For example, for ICMP, type numbers are used: the temporary entry specifies the same type number as the original packet (with only one exception: if the original ICMP packet is type 8, the returning ICMP packet must be type 0 to be matched).
- The entry inherits all the values of the original triggering packet, with exceptions only as noted in the previous four bullets.
- IP traffic entering your internal network will be evaluated against the entry, until the entry expires. If an IP packet matches the entry, the packet will be forwarded into your network.
- The entry will expire (be removed) after the last packet of the session is matched.
- If no packets belonging to the session are detected for a configurable length of time (the timeout period), the entry will expire.

Examples

The following example defines a reflexive access list *tcptraffic*, in an outbound access list that permits all Border Gateway Protocol and Enhanced Interior Gateway Routing Protocol traffic and denies all ICMP traffic. This example is for an external interface (an interface connecting to an external network).

First, the interface is defined and the access list is applied to the interface for outbound traffic.

```
interface Serial 1
 description Access to the Internet via this interface
 ip access-group outboundfilters out
```

Next, the outbound access list is defined and the reflexive access list *tcptraffic* is created with a reflexive **permit** entry.

```
ip access-list extended outboundfilters
 permit tcp any any reflect tcptraffic
```

Related Commands

Command	Description
evaluate	Nests a reflexive access list within an access list.
ip access-list	Defines an IP access list by name.
ip reflexive-list timeout	Specifies the length of time that reflexive access list entries will continue to exist when no packets in the session are detected.

permit (webvpn acl)

To set conditions to allow packets to pass a named Secure Sockets Layer Virtual Private Network (SSL VPN) access list, use the **permit** command in webvpn acl configuration mode. To remove a permit condition from an access list, use the **no** form of this command.

permit [**url** [**any** | *url-string*]] [**ip** | **tcp** | **udp** | **http** | **https** | **cifs**] [**any** | *source-ip source-mask*] [**any** | *destination-ip destination-mask*] **time-range** *time-range-name* [**syslog**]

no permit url [**any** | *url-string*] [**ip** | **tcp** | **udp** | **http** | **https** | **cifs**] [**any** | *source-ip source-mask*] [**any** | *destination-ip destination-mask*] **time-range** *time-range-name* [**syslog**]

Syntax Description

url	(Optional) Filtering rules are applied to a URL. • Use the **any** keyword as an abbreviation for any URL.
url-string	(Optional) URL string defined as follows: **scheme://host[:port][/path]** • **scheme**—Can be HTTP, Secure HTTPS (HTTPS), or Common Internet File System (CIFS). This field is required in the URL string. • **host**—Can be a hostname or a host IP (host mask). The host can have one wildcard (*). • **port**—Can be any valid port number (1–65535). It is possible to have multiple port numbers separated by a comma (,). The port range is expressed using a dash (-). • **path**—Can be any valid path string. In the path string, the $user is translated to the current user name.
ip	(Optional) Permits only IP packets. When you enter the **ip** keyword, you must use the specific command syntax shown for the IP form of the **permit** command.
tcp	(Optional) Permits only TCP packets. When you enter the **tcp** keyword, you must use the specific command syntax shown for the TCP form of the **permit** command.
udp	(Optional) Permitss only UDP packets. When you enter the **udp** keyword, you must use the specific command syntax shown for the UDP form of the **permit** command.
http	(Optional) Permits only HTTP packets. When you enter the **http** keyword, you must use the specific command syntax shown for the HTTP form of the **permit** command.
https	(Optional) Permits only HTTPS packets. When you enter the **https** keyword, you must use the specific command syntax shown for the HTTPS form of the **permit** command.
cifs	(Optional) Permits only CIFS packets. When you enter the **cifs** keyword, you must use the specific command syntax shown for the CIFS form of the **permit** command.

source-ip *source-mask*	(Optional) Number of the network or host from which the packet is being sent. There are three alternative ways to specify the source: • Use a 32-bit quantity in four-part dotted-decimal format. • Use the **any** keyword as an abbreviation for a source and source mask of 0.0.0.0 255.255.255.255. • Use **host** source as an abbreviation for a source and source-wildcard of source 0.0.0.0.
destination-ip *destination-mask*	(Optional) Number of the network or host to which the packet is being sent. There are three alternative ways to specify the destination: • Use a 32-bit quantity in four-part dotted-decimal format. • Use the **any** keyword as an abbreviation for a source and source mask of 0.0.0.0 255.255.255.255. • Use **host** source as an abbreviation for a source and source-wildcard of source 0.0.0.0.
time-range *time-range-name*	Name of the time range that applies to this **permit** statement. The name of the time range and its restrictions are specified by the **time-range** and **absolute** or **periodic** commands, respectively.
syslog	(Optional) System logging messages are generated.

Command Default

All packets are permitted.

Command Modes

Webvpn acl configuration

Command History

Release	Modification
12.4(11)T	This command was introduced.

Usage Guidelines

Use this command following the **acl** command (in webvpn context configuration mode) to specify conditions under which a packet can pass the named access list.

The **time-range** keyword allows you to identify a time range by name. The **time-range**, **absolute**, and **periodic** commands specify when this permit statement is in effect.

Examples

The following example shows that all packets from the URL "https://10.168.2.228:34,80-90,100-/public" are permitted to pass ACL "acl1":

```
webvpn context context1
 acl acl1
  permit url "https://10.168.2.228:34,80-90,100-/public"
```

Related Commands

Command	Description
absolute	Specifies an absolute time for a time range.
deny (webvpn acl)	Sets conditions in a named SSL VPN access list that will deny packets.
periodic	Specifies a recurring (weekly) time range for functions that support the time-range feature.
time-range	Enables time-range configuration mode and defines time ranges for extended access lists.

pfs

To configure a server to notify the client of the central-site policy regarding whether PFS is required for any IP Security (IPSec) Security Association (SA), use the **pfs** command in global configuration mode. To restore the default behavior, use the **no** form of this command.

pfs

no pfs

Syntax Description

This command has no arguments or keywords.

Defaults

The server will not notify the client of the central-site policy regarding whether PFS is required for any IPSec SA.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.3(4)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS 12.2SX family of releases. Support in a specific 12.2SX release is dependent on your feature set, platform, and platform hardware.

Usage Guidelines

Before you use the **pfs** command, you must first configure the **crypto isakmp client configuration group** command.

An example of an attribute-value (AV) pair for the PFS attribute is as follows:

```
ipsec:pfs=1
```

Examples

The following example shows that the server has been configured to notify the client of the central-site policy regarding whether PFS is required for any IPSec SA:

```
crypto isakmp client configuration group
 pfs
```

Related Commands

Command	Description
crypto isakmp client configuration group	Specifies to which group a policy profile will be defined.

pki-server

To specify the certificate server that is to be associated with the Trusted Transitive Introduction (TTI) exchange between the Secure Device Provisioning (SDP) petitioner and the SDP registrar, use the **pki-server** command in tti-registrar configuration mode. To change the specified certificate server, use the **no** form of this command.

pki-server *label*

no pki-server *label*

Syntax Description

label	Name of certificate server.

Defaults

A certificate server is not associated with the TTI exchange; thus, the petitioner and registrar will not be able to communicate.

Command Modes

tti-registrar configuration

Command History

Release	Modification
12.3(8)T	This command was introduced.

Usage Guidelines

Although any device that contains a crypto image can be the registrar, it is recommended that the registrar be either a Cisco IOS certificate server registration authority (RA) or a Cisco IOS certificate server root.

Examples

The following example shows how to associate the certificate server “cs1” with the TTI exchange:

```
crypto wui tti registrar
 pki-server cs1
```

Related Commands

Command	Description
crypto pki server	Enables a Cisco IOS certificate server and enters certificate server configuration mode.
crypto wui tti registrar	Configures a device to become an SDP registrar and enters tti-registrar configuration mode.

pki trustpoint

To use the PKI trustpoints in the Rivest, Shamir and Adleman (RSA) signature authentication method, use the **pki trustpoint** command in IKEv2 profile configuration mode. To remove the trustpoint, use the **no** form of this command.

pki trust-point *trustpoint-name* [**sign** | **verify**]

no pki trust-point *trustpoint-name* [**sign** | **verify**]

Syntax Description

trustpoint-name	The trustpoint name as defined in the global configuration.
sign	(Optional) Uses certificates from the trustpoint to create a digital signature that is sent to the peer.
verify	(Optional) Uses certificates from the trustpoint to validate digital signatures received from thepeer.

Command Default

If there is no trustpoint defined in the IKEv2 profile configuration, the default is to validate the certificate using all the trustpoints that are defined in the global configuration.

Command Modes

IKEv2 profile configuration (config-ikev2-profile)

Command History

Release	Modification
15.1(1)T	This command was introduced.
Cisco IOS XE Release 3.3S	This command was integrated into Cisco IOS XE Release 3.3S.

Usage Guidelines

The **pki trustpoint** command specifies the trustpoints that are used with the RSA-signature authentication method. You can configure upto six truspoints.

Note If the **sign** or **verify** keyword is not specified, the trustpoint is used for signing and verification.

Examples

The following example specifies two trustpoints, trustpoint-local for local authentication using sign and trustpoint-remote for remote verification using verify:

```
Router(config)# crypto ikev2 profile profile2
Router(config-ikev2-profile)# pki trustpoint trustpoint-local sign
Router(config-ikev2-profile)# pki trustpoint trustpoint-remote verify
```

Related Commands

Command	Description
crypto ikev2 profile	Defines an IKEv2 profile.

police (zone policy)

To limit traffic matching within a firewall (inspect) policy, use the **police** command in policy-map-class configuration mode. To remove traffic limiting from the firewall policy configuration, use the **no** form of this command.

police rate *bps* [**burst** *size*]

no police rate *bps* [**burst** *size*]

Syntax Description

rate *bps*	Average rate in bits per second (bps). Valid values are 8000 to 2000000000. **Note** Traffic limiting is in bps only; that is, packets per seconds (pps) and percent rates are not supported.
burst *size*	(Optional) Burst size in bytes. Valid values are 1000 to 51200000. The default normal burst size is 1500 bytes.

Command Default

Traffic limiting is disabled.

Command Modes

Policy-map-class configuration

Command History

Release	Modification
12.4(9)T	This command was introduced.

Usage Guidelines

Issue the **police** command within an inspect policy to limit the number of concurrent connections allowed for applications such as Instant Messenger (IM) and peer-to-peer (P2P).

To effectively use the **police** command, you must also enable Cisco IOS stateful packet inspection within the inspect policy map. If you configure the **police** command without configuring the inspect action (via the **inspect** command), you will receive an error message and the **police** command will be rejected.

Because an inspect policy map can be applied only to a zone pair, and not an interface, the police action will be enforced on traffic that traverses the zone pair. (The direction is inherent to the specification of the zone pair.)

The police action is not allowed in policies that are attached to zone pairs involving a “self” zone. If you want to perform this task, you should use control plane policing.

Examples

The following example shows how to limit traffic matching with the inspect policy “p1”:

```
policy-map type inspect p1
 class type inspect c1
  inspect
  police rate 1000 burst 6100
```

The following example is sample output from the s**how policy-map type inspect zone-pair** command, which can now be used to verify the police action configuration:

```
Router# show policy-map type inspect zone-pair

Zone-pair: zp

Service-policy inspect : test-udp

  Class-map: check-udp (match-all)
   Match: protocol udp
   Inspect
    Packet inspection statistics [process switch:fast switch]
    udp packets: [3:4454]

    Session creations since subsystem startup or last reset 92
    Current session counts (estab/half-open/terminating) [5:33:0]
    Maxever session counts (estab/half-open/terminating) [5:59:0]
    Last session created 00:00:06
    Last statistic reset never
    Last session creation rate 61
    Last half-open session total 33
   Police
   rate 8000 bps,1000 limit
   conformed 2327 packets, 139620 bytes; actions: transmit
   exceeded 36601 packets, 2196060 bytes; actions: drop
   conformed 6000 bps, exceed 61000 bps

  Class-map: class-default (match-any)
   Match: any
   Drop (default action)
   0 packets, 0 bytes
```

Related Commands

Command	Description
show policy-map type inspect zone-pair	Displays the runtime inspect type policy map statistics and other information such as sessions existing on a specified zone pair.

policy

To define the Central Policy Push (CPP) firewall policy push, use the command in global configuration mode. To remove the CPP policy that was configured, use the **no** form of this command.

policy {**check-presence** | **central-policy-push** {**access-list** {**in** | **out**} {*access-list-name* | *access-list-number*}}

no policy {**check-presence** | **central-policy-push** {**access-list** {**in** | **out**} {*access-list-name* | *access-list-number*}}

Syntax Description

check-presence	Instructs the server to check for the presence of the specified firewall as shown as *firewall-type* on the client.
central-policy-push	Pushes the CPP firewall policy push. The configuration following this keyword specifies the actual policy, such as the input and output access lists that have to be applied by the client firewall of the type *firewall-type*.
access-list in	Defines the inbound access list on the virtual private network (VPN) remote client.
access-list out	Defines the outbound access list on the VPN remote client.
access-list-name | *access-list-number*	Access list name or number.

Command Default

The CPP policy is not defined.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.4(6)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS release 12.(33)SRA.

Examples

The following example defines the CPP policy name as “hw-client-g-cpp.” The “Cisco-Security-Agent” policy type is mandatory. The CPP inbound list is “192” and the outbound list is “sample”:

```
crypto isakmp client firewall hw-client-g-cpp required Cisco-Security-Agent
 policy central-policy-push access-list in 192
 policy central-policy-push access-list out sample
 policy check-presence:
```

The following example shows access lists that have been applied on a VPN remote client and later applied by the client firewall :

Defines the inbound access control list that is applied on the VPN remote client

```
.
.
.
```

```
access-list 170 permit ip 172.18.124.0 0.0.0.255 any
access-list 170 permit ip 172.21.1.0 0.0.0.255 any
.
.
.
```

Defines the outbound ACL that is applied on the VPN remote client

```
.
.
.
access-list 180 permit ip any 172.18.124.0 0.0.0.255
.
.
.
```

Inbound and outbound policies to be applied by the client firewall

```
.
.
.
crypto isakmp client firewall test required cisco-integrated-client-firewall
  policy central-policy-push access-list in 170
  policy central-policy-push access-list out 180
.
.
.
crypto isakmp client configuration group vpngroup1
  firewall policy test
.
.
.
```

Related Commands

Command	Description
crypto isakmp client firewall	Defines the CPP) firewall push policy on a server.

policy group

To enter webvpn group policy configuration mode to configure a group policy, use the **policy group** command in webvpn context configuration mode. To remove the policy group from the router configuration file, use the **no** form of this command.

policy group *name*

no policy group *name*

Syntax Description

name	Name of the policy group.

Command Default

Webvpn group policy configuration mode is not entered, and a policy group is not configured.

Command Modes

Webvpn context configuration

Command History

Release	Modification
12.4(6)T	This command was introduced.

Usage Guidelines

The policy group is a container that defines the presentation of the portal and the permissions for resources that are configured for a group of end users. Entering the **policy group** command places the router in webvpn group policy configuration mode. After the group policy is configured, the policy group is attached to the SSL VPN context configuration by configuring the **default-group-policy** command.

Examples

The following example configures a policy group named ONE:

```
Router(config)# webvpn context context1
Router(config-webvpn-context)# policy group ONE
Router(config-webvpn-group)# exit
Router(config-webvpn-context)# default-group-policy ONE
```

Related Commands

Command	Description
banner	Configures a banner to be displayed after a successful login.
citrix enabled	Enables Citrix application support for end users in a policy group.
default-group-policy	Configures a default group policy for SSL VPN sessions.
filter citrix	Configures a Citrix application access filter.
filter tunnel	Configures a SSL VPN tunnel access filter.
functions	Enables a file access function or tunnel mode support in a group policy configuration.
hide-url-bar	Prevents the URL bar from being displayed on the SSL VPN portal page.

Command	Description
nbns-list (policy group)	Attaches a NBNS server list to a policy group configuration.
port-forward (policy group)	Attaches a port-forwarding list to a policy group configuration.
svc address-pool	Configures a pool of IP addresses to assign to end users in a policy group.
svc default-domain	Configures the domain for a policy group.
svc dns-server	Configures DNS servers for policy group end users.
svc dpd-interval	Configures the DPD timer value for the gateway or client.
svc homepage	Configures the URL of the web page that is displayed upon successful user login.
svc keep-client-installed	Configures the end user to keep Cisco AnyConnect VPN Client software installed when the SSL VPN connection is not enabled.
svc msie-proxy	Configures MSIE browser proxy settings for policy group end users.
svc msie-proxy server	Specifies a Microsoft Internet Explorer proxy server for policy group end users.
svc rekey	Configures the time and method that a tunnel key is refreshed for policy group end users.
svc split	Configures split tunneling for policy group end users.
svc wins-server	Configures configure WINS servers for policy group end users.
timeout	Configures the length of time that an end user session can remain idle or the total length of time that the session can remain connected.
url-list (policy group)	Attaches a URL list to policy group configuration.
webvpn context	Enters webvpn context configuration mode to configure the SSL VPN context.

policy-map type inspect

To create a Layer 3 and Layer 4 or a Layer 7 (protocol-specific) inspect type policy map, use the **policy-map type inspect** command in global configuration mode. To delete an inspect type policy map, use the **no** form of this command.

Layer 3 and Layer 4 (Top Level) Policy Map Syntax

policy-map type inspect *policy-map-name*

no policy-map type inspect *policy-map-name*

Layer 7 (Application-Specific) Policy Map Syntax

policy-map type inspect *protocol-name policy-map-name*

no policy-map type inspect *protocol-name policy-map-name*

Syntax Description

policy-map-name	Name of the policy map. The name can be a maximum of 40 alphanumeric characters.
protocol-name	Layer 7 application-specific policy map. The supported protocols are as follows: • **h323**—H.323 protocol, Version 4 • **http**—HTTP • **im**—Instant Messenger (IM) protocol For **im**, the supported IM protocols include: – AOL Version 5 and later versions – I Seek You (ICQ) Version 2003b.5.56.1.3916.85 – MSN Messenger Version 6.x and 7.x – Windows Messenger Version 5.1.0701 – Yahoo Messenger Version 9.0 and later versions • **imap**—Internet Message Access Protocol (IMAP) • **p2p**—Peer-to-peer (P2P) protocol • **pop3**—Post Office Protocol, Version 3 (POP3) • **sip**—Session Initiation Protocol (SIP) • **smtp**—Simple Mail Transfer Protocol (SMTP) • **sunrpc**—Sun Remote Procedure Call (SUNRPC)

Command Default

No policy-map is configured.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.4(6)T	This command was introduced.
12.4(9)T	Support for the following protocols and keywords was added: • P2P protocol and the **p2p** keyword • IM protocol and the **im** keyword
12.4(15)XZ	This command was integrated into Cisco IOS Release 12.4(15)XZ. Support for the SIP protocol was added.
12.4(20)T	Support for the ICQ and Windows Messenger IM protocols and following keywords was added: **icq**, **winmsgr** Support for the H.323 VoIP protocol and following keyword was added: **h323**

Usage Guidelines

Use the **policy-map type inspect** command to create a Layer 3, Layer 4 inspect type policy map or a Layer 7 application-specific inspect type policy map. After you create a policy map, you should enter the **class type inspect** command (as appropriate for your configuration) to specify the traffic (class) on which an action is to be performed. The class was previously defined in a class map. Thereafter, you should enter the **inspect** command to enable Cisco IOS stateful packet inspection and to specify inspect-specific parameters in a parameter map.

Layer 3, Layer 4 (Top Level) Policy Maps

Top-level policy maps allow you to define high-level actions such as **inspect**, **drop**, **pass**, and **urlfilter**. You can attach the maps to a target (zone pair). The maps can contain "child" policies that are also known as application-specific Layer 7 policies.

Layer 7 (Application-Specific) Policy Maps

Application-specific policy maps are used to specify a policy for an application protocol. For example, if you want to drop HTTP traffic with Uniform Resource Identifier (URI) lengths exceeding 256 bytes, you must configure an HTTP policy map to do that. Application-specific policy maps cannot be attached directly to a target (zone pair). They must be configured as "child" policies in a top-level Layer 3 or Layer 4 policy map.

Examples

The following example specifies the traffic class (host) on which the drop action is to be performed:

```
policy-map type inspect mypolicy
 class type inspect host
 drop
```

The following example shows how to configure the policy map "my-im-pmap" with two IM classes—AOL and Yahoo Messenger—and allow only text-chat messages to pass through. When any packet with a service other than "text-chat" is seen, the connection will be reset.

```
class-map type inspect aol match-any my-aol-cmap
 match service text-chat
!
class-map type inspect ymsgr match-any my-ysmgr-cmap
 match service any
!
policy-map type inspect im my-im-pmap
 class type inspect aol my-aol-cmap
 allow
 log
!
 class type inspect ymsgr my-ysmgr-cmap
 reset
 log
```

Related Commands

Command	Description
class type inspect	Specifies the traffic (class) on which an action is to be performed.

policy-map type inspect urlfilter

To create or modify a URL filter type inspect policy map, use the **policy-map type inspect urlfilter** command in global configuration mode. To delete a URL filter type inspect policy map, use the **no** form of this command.

policy-map type inspect urlfilter *policy-map-name*

no policy-map type inspect urlfilter *policy-map-name*

Syntax Description

policy-map-name	Name of the policy map.

Command Default

No policy map is created.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.4(15)XZ	This command was introduced.
12.4(20)T	This command was integrated into Cisco IOS Release 12.4(20)T.

Usage Guidelines

Use the **policy-map type inspect urlfilter** command to create a URL filter type inspect policy map. The policy map specifies the traffic (**class type urlfilter**) and the actions to be performed on that traffic for the specified URL filtering policy.

Before you create a URL filter type inspect policy map, use the following commands:

- **class-map type urlfilter** command to configure the match criteria for the traffic.
- **parameter-map type urlfpolicy** command to specify the parameters for the URL filtering server. If you are configuring a policy for a Trend Router Provisioning Server (TRPS), you must also specify the global filtering parameters with the **parameter-map type trend-global** command.

After you create a policy map, use the following commands to configure the URL filtering policy:

- **class type urlfilter** [**trend** | **n2h2** | **websense**] *class-name*—Specifies the class of traffic to which the policy applies. If you specify an optional URL filtering server, you must also use the **parameter type urlfpolicy** command to specify the appropriate per-policy parameters for that URL filtering server.

  For each class, use one of the URL filtering action commands to specify how to handle a URL that matches the class map. Table 53 lists the URL filtering action commands.

  Table 53 URL Filtering Action Commands

Command	Description
allow	Permits access to the requested URL.
log	Logs the URL request.

Table 53 URL Filtering Action Commands

Command	Description
reset	Resets the HTTP connection at both ends.
server-specified action	Specifies that the traffic is handled by the URL filtering server. This action is valid only for Websense and N2H2 classes.

- **description** *string*—Describes the policy.
- **exit**—Exits the policy map.
- **no**—Negates or sets the default value for a command.
- **parameter type urlfpolicy** [**trend** | **n2h2** | **websense**]—Specifies what type of URL filtering this policy applies to: local (default), Trend Micro, SmartFilter, or Websense.
- **rename** *policy-map-name*—Specifies a new name for the policy map.

Examples

The following example shows a how to create a URL filter type inspect policy for a Trend Micro URL filtering server. The policy logs URL requests that match the URL categories specified in the class drop-category, and then resets the connection, thus denying the request.

```
class-map type urlfilter trend match-any drop-category
 match url category Gambling
 match url category Personals-Dating

parameter-map type trend-global global-parameter-map
 server trend.example.com

parameter-map type urlfpolicy trend g1-trend-pm
 max-request 2147483647
 max-resp-pak 20000
 allow-mode on
 truncate hostname
 block-page message "group1: 10.10.10.0 is blocked by Trend."

policy-map type inspect urlfilter g1-trend-policy
 parameter type urlfpolicy trend g1-trend-parameter-map
 class type urlfilter trend drop-category
  log
 reset
```

The following example shows a filtering policy for a Websense URL filtering server. The policy logs and allows URL requests that are in the trusted domain class, logs and denies URL requests that are in the untrusted domain class, and logs and denies URL requests that are in the keyword class.

```
policy-map type inspect urlfilter websense-policy
 parameter type urlfpolicy websense websense-parameter-map
 class type urlfilter trusted-domain-class
  log
  allow
 class type urlfilter untrusted-domain-class
  log
  reset
 class type urlfilter keyword-class
  log
  reset
```

Related Commands

Command	Description
class-map type urlfilter	Specifies the class on which a policy action is to be performed.
class type urlfilter	Associates a URL filter class map with a URL filtering policy maps.
parameter-map type trend-global	Creates or modifies the parameter map for global TRPS parameters.
parameter-map type urlfpolicy	Creates or modifies a parameter map for a URL filtering policy.

pool (isakmp-group)

To define a local pool address, use the **pool** command in ISAKMP group configuration mode or IKEv2 authorization policy configuration mode. To remove a local pool from your configuration, use the **no** form of this command.

pool *name*

no pool *name*

Syntax Description

name	Name of the local address pool.

Defaults

No local pool address is defined.

Command Modes

ISAKMP group configuration (config-isakmp-group)
IKEv2 authorization policy configuration (config-ikev2-author-policy)

Command History

Release	Modification
12.2(8)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS 12.2SX family of releases. Support in a specific 12.2SX release is dependent on your feature set, platform, and platform hardware.
Cisco IOS XE Release 3.3S	This command was integrated into Cisco IOS XE Release 3.3S.

Usage Guidelines

Use the **pool** command to refer to an IP local pool address, which defines a range of addresses that will be used to allocate an internal IP address to a client. Although a user must define at least one pool name, a separate pool may be defined for each group policy.

Note This command must be defined and refer to a valid IP local pool address, or the client connection will fail.

You must enable the following commands before enabling the **dns** command:

- **crypto isakmp client configuration group**—Specifies the group policy information that has to be defined or changed.
- **crypto ikev2 authorization policy**—Specifies the local group policy authorization parameters.

Examples

The following example shows how to refer to the local pool address “dog”:

```
crypto isakmp client configuration group cisco
```

```
  key cisco
  dns 10.2.2.2 10.3.2.3
  pool dog
  acl 199
!
ip local pool dog 10.1.1.1 10.1.1.254
```

Related Commands

Command	Description
acl	Configures split tunneling.
crypto ikev2 authorization policy	Specifies an IKEv2 authorization policy group.
crypto isakmp client configuration group	Specifies the DNS domain to which a group belongs.
ip local pool	Configures a local pool of IP addresses to be used when a remote peer connects to a point-to-point interface.

port

To specify the port on which a device listens for RADIUS requests from configured RADIUS clients, use the **port** command in dynamic authorization local server configuration mode. To restore the default, use the **no** form of this command.

port *port-number*

no port *port-number*

Syntax Description

port-number	Port number. The default value is port 1700.

Command Default

The device listens for RADIUS requests on the default port (port 1700).

Command Modes

Dynamic authorization local server configuration (config-locsvr-da-radius)

Command History

Release	Modification
12.2(28)SB	This command was introduced.
Cisco IOS XE Release 2.6	This command was integrated into Cisco IOS XE Release 2.6.

Usage Guidelines

A device (such as a router) can be configured to allow an external policy server to dynamically send updates to the router. This functionality is facilitated by the CoA RADIUS extension. CoA introduced peer-to-peer capability to RADIUS, enabling a router and external policy server each to act as a RADIUS client and server. Use the **port** command to specify the ports on which the router will listen for requests from RADIUS clients.

Examples

The following example specifies port 1650 as the port on which the device listens for RADIUS requests:

```
aaa server radius dynamic-author
 client 10.0.0.1
 port 1650
```

Related Commands

Command	Description
aaa server radius dynamic-author	Configures a device as a AAA server to facilitate interaction with an external policy server.

port-forward

To enter webvpn port-forward list configuration mode to configure a port-forwarding list, use the **port-forward** command in webvpn context configuration mode. To remove the port-forwarding list from the SSL VPN context configuration, use the **no** form of this command.

port-forward *name*

no port-forward *name*

Syntax Description

name	Name of the port-forwarding list.

Command Default

Webvpn port-forward list configuration mode is not entered, and a port-forwarding list is not configured.

Command Modes

Webvpn context configuration

Command History

Release	Modification
12.3(14)T	This command was introduced.

Usage Guidelines

The **port-forward** command is used to create the port-forwarding list. Application port number mapping (port forwarding) is configured with the **local-port** command in webvpn port-forward configuration mode.

A port-forwarding list is configured for thin client mode SSL VPN. Port forwarding extends the cryptographic functions of the SSL-protected browser to provide remote access to TCP-based applications that use well-known port numbers, such as POP3, SMTP, IMAP, Telnet, and SSH.

When port forwarding is enabled, the hosts file on the SSL VPN client is modified to map the application to the port number configured in the forwarding list. The application port mapping is restored to default when the user terminates the SSL VPN session.

Examples

The following example configures port forwarding for well-known e-mail application port numbers:

```
Router(config)# webvpn context context1
Router(config-webvpn-context)# port-forward EMAIL
Router(config-webvpn-port-fwd)# local-port 30016 remote-server mail.company.com
remote-port 110 description POP3
Router(config-webvpn-port-fwd)# local-port 30017 remote-server mail.company.com
remote-port 25 description SMTP
Router(config-webvpn-port-fwd)# local-port 30018 remote-server mail.company.com
remote-port 143 description IMAP
```

Related Commands

Command	Description
local-port (WebVPN)	Remaps an application port number in a port-forwarding list.
webvpn context	Enters webvpn context configuration mode to configure the SSL VPN context.

port-forward (policy group)

To attach a port-forwarding list to a policy group configuration, use the **port-forward** command in webvpn group policy configuration mode. To remove the port-forwarding list from the policy group configuration, use the **no** form of this command.

port-forward *name* [**auto-download** [**http-proxy** [**proxy-url** *homepage-url*]] | **http-proxy** [**proxy-url** *homepage-url*] [**auto-download**]]

no port-forward *name* [**auto-download** [**http-proxy** [**proxy-url** *homepage-url*]] | **http-proxy** [**proxy-url** *homepage-url*] [**auto-download**]]

Syntax Description

name	Name of the port-forwarding list that was configured in webvpn context configuration mode.
auto-download	(Optional) Allows for automatic download of the port-forwarding Java applet on the portal page of a website.
http-proxy	(Optional) Allows the Java applet to act as a proxy for the browser of the user.
proxy-url *homepage-url*	(Optional) Page at this URL address opens as the portal page of the user.

Command Default

A port-forwarding list is not attached to a policy group configuration.

Command Modes

Webvpn group policy configuration (config-webvpn-group)

Command History

Release	Modification
12.4(6)T	This command was introduced.
12.4(9)T	This command was modified. The **auto-download** keyword was added.

Usage Guidelines

The configuration of this command applies to only clientless access mode. In clientless mode, the remote user accesses the internal or corporate network using the web browser on the client machine.

Examples

The following example shows how to apply the port-forwarding list to the policy group configuration:

```
webvpn context context1
 port-forward EMAIL
  local-port 30016 remote-server mail.company.com remote-port 110 description POP3
  local-port 30017 remote-server mail.company.com remote-port 25 description SMTP
  local-port 30018 remote-server mail.company.com remote-port 143 description IMAP
  exit
 policy group ONE
 port-forward EMAIL auto-download
```

The following example shows that HTTP proxy has been configured. The page at URL "http://www.example.com" will automatically download as the home page of the user.

```
webvpn context myContext
 ssl authenticate verify all
 !
 !
 port-forward "email"
   local-port 20016 remote-server "ssl-server1.sslvpn-ios.com" remote-port 110 description
"POP-ssl-server1"
 !
 policy group myPolicy
   port-forward "email" auto-download http-proxy proxy-url "http://www.example.com"
 inservice
```

Related Commands

Command	Description
local-port (WebVPN)	Remaps an application port number in a port-forwarding list.
policy group	Enters webvpn group policy configuration mode to configure a group policy.
port-forward	Enters webvpn port-forward list configuration mode to configure a port-forwarding list.
webvpn context	Enters webvpn context configuration mode to configure the SSL VPN context.

port-misuse

To permit or deny HTTP traffic through the firewall on the basis of specified applications in the HTTP message, use the **port-misuse** command in appfw-policy-http configuration mode. To disable this inspection parameter, use the **no** form of this command.

port-misuse {**p2p** | **tunneling** | **im** | **default**} **action** {**reset** | **allow**} [**alarm**]

no port-misuse {**p2p** | **tunneling** | **im** | **default**} **action** {**reset** | **allow**} [**alarm**]

Syntax Description

p2p	Peer-to-peer protocol applications subject to inspection: Kazaa and Gnutella.
tunneling	Tunneling applications subject to inspection: HTTPPort/HTTPHost, GNU Httptunnel, GotoMyPC, Firethru, Http-tunnel.com Client
im	Instant messaging protocol applications subject to inspection: Yahoo Messenger.
default	All applications are subject to inspection.
action	Applications detected within the HTTP messages that are outside of the specified application are subject to the specified action (**reset** or **allow**).
reset	Sends a TCP reset notification to the client or server if the HTTP message fails the mode inspection.
allow	Forwards the packet through the firewall.
alarm	(Optional) Generates system logging (syslog) messages for the given action.

Defaults

If this command is not enabled, HTTP messages are permitted through the firewall if any of the applications are detected within the message.

Command Modes

appfw-policy-http configuration

Command History

Release	Modification
12.3(14)T	This command was introduced.

Examples

The following example shows how to define the HTTP application firewall policy “mypolicy.” This policy includes all supported HTTP policy rules. After the policy is defined, it is applied to the inspection rule “firewall,” which will inspect all HTTP traffic entering the FastEthernet0/0 interface.

```
! Define the HTTP policy.
appfw policy-name mypolicy
 application http
  strict-http action allow alarm
  content-length maximum 1 action allow alarm
  content-type-verification match-req-rsp action allow alarm
  max-header-length request 1 response 1 action allow alarm
  max-uri-length 1 action allow alarm
  port-misuse default action allow alarm
  request-method rfc default action allow alarm
  request-method extension default action allow alarm
  transfer-encoding type default action allow alarm
!
!
! Apply the policy to an inspection rule.
ip inspect name firewall appfw mypolicy
ip inspect name firewall http
!
!
! Apply the inspection rule to all HTTP traffic entering the FastEthernet0/0 interface.
interface FastEthernet0/0
 ip inspect firewall in
!
!
```