# mab

To enable MAC-based authentication on a port, use the **mab** command in interface configuration mode. To disable MAC-based authentication, use the **no** form of this command.

**mab** [**eap**]

**no mab**

## Syntax Description

| | |
|---|---|
| **eap** | (Optional) Configures the port to use Extensible Authentication Protocol (EAP). |

## Command Default

MAC-based authentication is not enabled.

## Command Modes

Interface configuration (config-if)

## Command History

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

## Usage Guidelines

Use the **mab** command to enable MAC-based authentication on a port. To enable EAP on the port, use the **mab eap** command.

**Note** If you are unsure whether MAB or MAB EAP is enabled or disabled on the switched port, use the **default mab** or **default mab eap** commands in interface configuration mode to configure MAB or MAB EAP to its default.

## Examples

The following example shows how to configure MAC-based authorization on a Gigabit Ethernet port:

```
Switch(config)# interface GigabitEthernet6/2
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config-if)# mab
Switch(config-if)# end
```

## Related Commands

| Command | Description |
|---|---|
| **show mab** | Displays information about MAB. |

# mac access-group

To use a MAC access control list (ACL) to control the reception of incoming traffic on a Gigabit Ethernet interface, an 802.1Q VLAN subinterface, an 802.1Q-in-Q stacked VLAN subinterface, use the **mac access-group** command in interface or subinterface configuration mode. To remove a MAC ACL, use the **no** form of this command.

**mac access-group** *access-list-number* **in**

**no mac access-group** *access-list-number* **in**

## Syntax Description

| | |
|---|---|
| *access-list-number* | Number of a MAC ACL to apply to an interface or subinterface (as specified by a **access-list (MAC)** command). This is a decimal number from 700 to 799. |
| **in** | Filters on inbound packets. |

## Defaults

No access list is applied to the interface or subinterface.

## Command Modes

Interface configuration (config-if)
Subinterface configuration (config-subif)

## Command History

| Release | Modification |
|---|---|
| 12.0(32)S | This command was introduced on the Cisco 12000 series Internet router. |
| 12.2(33)SXH | This command was integrated into Cisco IOS Release 12.2(33)SXH. |

## Usage Guidelines

MAC ACLs are applied on incoming traffic on Gigabit Ethernet interfaces and VLAN subinterfaces. After a networking device receives a packet, the Cisco IOS software checks the source MAC address of the Gigabit Ethernet, 802.1Q VLAN, or 802.1Q-in-Q packet against the access list. If the MAC access list permits the address, the software continues to process the packet. If the access list denies the address, the software discards the packet and returns an Internet Control Message Protocol (ICMP) host unreachable message.

If the specified MAC ACL does not exist on the interface or subinterface, all packets are passed.

On Catalyst 6500 series switches, this command is supported on Layer 2 ports only.

**Note** The **mac access-group** command is supported on a VLAN subinterface only if a VLAN is already configured on the subinterface.

## Examples

The following example applies MAC ACL 101 on incoming traffic received on Gigabit Ethernet interface 0:

```
Router> enable
Router# configure terminal
```

```
Router(config)# interface gigabitethernet 0
Router(config-if)# mac access-group 101 in
```

## Related Commands

| Command | Description |
|---|---|
| **access-list (MAC)** | Defines a MAC ACL. |
| **clear mac access-list counters** | Clears the counters of a MAC ACL. |
| **ip access-group** | Configures an IP access list to be used for packets transmitted from the asynchronous host. |
| **show access-group mode interface** | Displays the ACL configuration on a Layer 2 interface. |
| **show mac access-list** | Displays the contents of one or all MAC ACLs. |

# mac-address (RITE)

To specify the Ethernet address of the destination host, use the **mac-address** command in router IP traffic export (RITE) configuration mode. To change the MAC address of the destination host, use the **no** form of this command.

**mac-address** *H.H.H*

**no mac-address** *H.H.H*

## Syntax Description

| | |
|---|---|
| *H.H.H* | 48-bit MAC address. |

## Defaults

A destination host is not known.

## Command Modes

RITE configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(4)T | This command was introduced. |
| 12.2(25)S | This command was integrated into Cisco IOS Release 12.2(25)S. |

## Usage Guidelines

The **mac-address** command, which is used to specify the destination host that is receiving the exported traffic, is part of suite of RITE configuration mode commands that are used to control various attributes for both incoming and outgoing IP traffic export.

The **ip traffic-export profile** command allows you to begin a profile that can be configured to export IP packets as they arrive or leave a selected router ingress interface. A designated egress interface exports the captured IP packets out of the router. Thus, the router can export unaltered IP packets to a directly connected device.

## Examples

The following example shows how to configure the profile "corp1," which will send captured IP traffic to host "00a.8aab.90a0" at the interface "FastEthernet 0/1." This profile is also configured to export one in every 50 packets and to allow incoming traffic only from the access control lists (ACL) "ham_ACL."

```
Router(config)# ip traffic-export profile corp1
Router(config-rite)# interface FastEthernet 0/1
Router(config-rite)# bidirectional
Router(config-rite)# mac-address 00a.8aab.90a0
Router(config-rite)# outgoing sample one-in-every 50
Router(config-rite)# incoming access-list ham_acl
Router(config-rite)# exit
Router(config)# interface FastEthernet 0/0
Router(config-if)# ip traffic-export apply corp1
```

**Related Commands**

| Command | Description |
|---|---|
| **ip traffic-export profile** | Creates or edits an IP traffic export profile and enables the profile on an ingress interface. |

# map type

To define the mapping of an attribute in the Lightweight Directory Access Protocol (LDAP) server, use the **map type** command in attribute-map configuration mode. To remove the attribute maps, use the **no** form of this command.

**map type** *ldap-attr-type aaa-attr-type* [**format** *dn-to-string*]

**no map type** *ldap-attr-type aaa-attr-type* [**format** *dn-to-string*]

## Syntax Description

| | |
|---|---|
| *ldap-attr-type* | LDAP attribute type. |
| *aaa-attr-type* | Authentication, Authorization, and Accounting (AAA) attribute type. |
| **format** | (Optional) Specifies the format conversion for attribute. |
| *dn-to-string* | (Optional) Converts the distinguished name (DN) to string format. |

## Command Default

No mapping types are defined.

## Command Modes

Attribute-map configuration (config-attr-map)

## Command History

| Release | Modification |
|---|---|
| 15.1(1)T | This command was introduced. |

## Usage Guidelines

To use the attribute mapping features, you need to understand the Cisco AAA attribute names and values as well as the LDAP servers user-defined attribute names and values.

## Examples

The following example shows how to map the user-defined attribute named department to the AAA attribute named element-req-qos in an LDAP server.

```
Router(config)# ldap attribute-map att_map_1
Router(config-attribute-map)# map type department element-req-qos format dn-to-string
Router(config-attribute-map)# exit
```

## Related Commands

| Command | Description |
|---|---|
| **attribute-map** | Attaches an attribute map to a particular LDAP server. |
| **ldap attribute-map** | Configures a dynamic LDAP attribute map. |
| **map-type** | Defines the mapping of a attribute in the LDAP server. |
| **show ldap attribute** | Displays information about default LDAP attribute mapping. |

# mask (policy-map)

To explicitly mask specified SMTP commands or the parameters returned by the server in response to an EHLO command, use the **mask** command in global configuration mode. To remove this filter from the configuration, use the **no** form of this command:

**mask**

**no mask**

## Command Default

The command-level default is not enabled.

## Command Modes

Policy-map configuration mode.

## Command History

| Release | Modification |
|---|---|
| 12.4(20)T | This command was introduced. |

## Usage Guidelines

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

## Examples

The following example shows how the **mask** command is used with the **match cmd** command and **verb** keyword to prevent ESMTP inspection:

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

policy-map type inspect smtp c1
    class type inspect smtp c1
        mask
```

## Related Commands

| Command | Description |
|---|---|
| **match cmd** | Specifies a value that limits the length of the ESMTP command line or the ESMTP command line verb used to thwart denial of service (DoS) attacks |

# mask-urls

To obfuscate, or mask, sensitive portions of an enterprise URL, such as IP addresses, hostnames, or port numbers, use the **mask-urls** command in webvpn group policy configuration mode. To remove the masking, use the **no** form of this command.

**mask-urls**

**no mask-urls**

## Syntax Description

This command has no arguments or keywords.

## Command Default

Sensitive portions of an enterprise URL are not masked.

## Command Modes

Webvpn group policy configuration

## Command History

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

## Usage Guidelines

This command is configured in group configuration only.

## Examples

The following example shows that URL obfuscation (masking) has been configured for policy group "GP":

```
Router(config)# webvpn context context1
Router(config-webvpn-context)# policy group GP
Router(config-webvpn-group)# mask-urls
```

## Related Commands

| Command | Description |
|---|---|
| **policy group** | Enters webvpn group policy configuration mode to configure a policy group. |
| **webvpn context** | Enters webvpn context configuration mode to configure the SSL VPN context. |

# match access-group

To configure the match criteria for a class map on the basis of the specified access control list (ACL), use the **match access-group** command in class-map configuration mode. To remove ACL match criteria from a class map, use the **no** form of this command.

**match access-group** {*access-group* | **name** *access-group-name*}

**no match access-group** *access-group*

## Syntax Description

| | |
|---|---|
| *access-group* | Numbered ACL whose contents are used as the match criteria against which packets are checked to determine if they belong to this class. An ACL number can be a number from 1 to 2699. |
| **name** *access-group-name* | Named ACL whose contents are used as the match criteria against which packets are checked to determine if they belong to this class. The name can be a maximum of 40 alphanumeric characters. |

## Command Default

No match criteria are configured.

## Command Modes

Class-map configuration

## Command History

| Release | Modification |
|---|---|
| 12.0(5)T | This command was introduced. |
| 12.0(5)XE | This command was integrated into Cisco IOS Release 12.0(5)XE. |
| 12.0(7)S | This command was integrated into Cisco IOS Release 12.0(7)S. |
| 12.0(17)SL | This command was enhanced to include matching on access lists on the Cisco 10000 series router. |
| 12.1(1)E | This command was integrated into Cisco IOS Release 12.1(1)E. |
| 12.4(6)T | This command was enhanced to support Zone-Based Policy Firewall. |
| 12.2(33)SRA | This command was integrated into Cisco IOS Release 12.2(33)SRA. |
| 12.2(31)SB | This command was integrated into Cisco IOS Release 12.2(31)SB. |
| 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

For class-based weighted fair queueing (CBWFQ), you define traffic classes based on match criteria including ACLs, protocols, input interfaces, quality of service (QoS) labels, and EXP field values. Packets satisfying the match criteria for a class constitute the traffic for that class.

**Note** For Zone-Based Policy Firewall, this command is not applicable to CBWFQ.

The **match access-group** command specifies a numbered or named ACL whose contents are used as the match criteria against which packets are checked to determine if they belong to the class specified by the class map.

When packets are matched to an access group, a traffic rate is generated for these packets. In a zone-based firewall policy, only the first packet that creates a session matches the policy. Subsequent packets in this flow do not match the filters in the configured policy, but instead match the session directly. The statistics related to subsequent packets are shown as part of the 'inspect' action.

#### Supported Platforms Other than Cisco 10000 Series Routers

To use the **match access-group** command, you must first enter the **class-map** command to specify the name of the class whose match criteria you want to establish. After you identify the class, you can use one of the following commands to configure its match criteria:

- **match access-group**
- **match input-interface**
- **match mpls experimental**
- **match protocol**

**Note** Zone-Based Policy Firewall supports only the **match access-group**, **match protocol**, and **match class-map** commands.

If you specify more than one command in a class map, only the last command entered applies. The last command overrides the previously entered commands.

**Note** The **match access-group** command specifies the numbered access list against whose contents packets are checked to determine if they match the criteria specified in the class map. Access lists configured with the optional **log** keyword of the **access-list** command are not supported when you configure match criteria. For more information about the **access-list** command, refer to the *Cisco IOS IP Application Services Command Reference*.

#### Cisco 10000 Series Routers

To use the **match access-group** command, you must first enter the **class-map** command to specify the name of the class whose match criteria you want to establish.

**Note** The **match access-group** command specifies the numbered access list against whose contents packets are checked to determine if they match the criteria specified in the class map. Access lists configured with the optional **log** keyword of the **access-list** command are not supported when you configure match criteria.

The following example specifies a class map called acl144 and configures the ACL numbered 144 to be used as the match criterion for that class:

```
class-map acl144
 match access-group 144
```

The following example pertains to Zone-Based Policy Firewall. The example defines a class map called c1 and configures the ACL numbered 144 to be used as the match criterion for that class.

```
class-map type inspect match-all c1
 match access-group 144
```

**Related Commands**

| Command | Description |
|---|---|
| **access-list (IP extended)** | Defines an extended IP access list. |
| **access-list (IP standard)** | Defines a standard IP access list. |
| **class-map** | Creates a class map to be used for matching packets to a specified class. |
| **match input-interface** | Configures a class map to use the specified input interface as a match criterion. |
| **match mpls experimental** | Configures a class map to use the specified EXP field value as a match criterion. |
| **match protocol** | Configures the match criteria for a class map on the basis of the specified protocol. |

# match address (GDOI local server)

To specify an IP extended access list for a Group Domain of Interpretation (GDOI) registration, use the **match address** command in GDOI SA IPsec configuration mode. To disable the access list, use the **no** form of this command.

**match address** {**ipv4** *access-list-number* | *access-list-name*}

**no match address** {**ipv4** *access-list-number* | *access-list-name*}

## Syntax Description

| | |
|---|---|
| **ipv4** | Specifies that IPv4 packets should be matched. |
| *access-list-number* \| *access-list-name* | Access list number or name. This value should match the access-list number or name of the extended access list that is being matched.<br>The range is 100 through 199 or 2000 through 2699 for an expanded range. |

## Command Default

No access lists are matched to the GDOI entry.

## Command Modes

GDOI SA IPsec configuration (gdoi-sa-ipsec)

## Command History

| Release | Modification |
|---|---|
| 12.4(6)T | This command was introduced. |
| Cisco IOS XE Release 2.3 | This command was implemented on the Cisco ASR 1000 series routers. |

## Examples

The following example shows that the IP extended access list is 102:

```
match address ipv4 102
```

## Related Commands

| Command | Description |
|---|---|
| **crypto gdoi group** | Identifies a GDOI group and enters GDOI group configuration mode. |
| **server local** | Designates a device as a GDOI key server and enters GDOI local server configuration. |

# match address (IPSec)

To specify an extended access list for a crypto map entry, use the **match address** command in crypto map configuration mode. To remove the extended access list from a crypto map entry, use the **no** form of this command.

**match address** [*access-list-id* | *name*]

**no match address** [*access-list-id* | *name*]

## Syntax Description

| | |
|---|---|
| *access-list-id* | (Optional) Identifies the extended access list by its name or number. This value should match the *access-list-number* or *name* argument of the extended access list being matched. |
| *name* | (Optional) Identifies the named encryption access list. This name should match the *name* argument of the named encryption access list being matched. |

## Defaults

No access lists are matched to the crypto map entry.

## Command Modes

Crypto map configuration

## Command History

| Release | Modification |
|---|---|
| 11.2 | 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 required for all static crypto map entries. If you are defining a dynamic crypto map entry (with the **crypto dynamic-map** command), this command is not required but is strongly recommended.

Use this command to assign an extended access list to a crypto map entry. You also need to define this access list using the **access-list** or **ip access-list extended** commands.

The extended access list specified with this command will be used by IPSec to determine which traffic should be protected by crypto and which traffic does not need crypto protection. (Traffic that is permitted by the access list will be protected. Traffic that is denied by the access list will not be protected in the context of the corresponding crypto map entry.)

Note that the crypto access list is *not* used to determine whether to permit or deny traffic through the interface. An access list applied directly to the interface makes that determination.

The crypto access list specified by this command is used when evaluating both inbound and outbound traffic. Outbound traffic is evaluated against the crypto access lists specified by the interface's crypto map entries to determine if it should be protected by crypto and if so (if traffic matches a **permit** entry) which crypto policy applies. (If necessary, in the case of static IPSec crypto maps, new security

associations are established using the data flow identity as specified in the **permit** entry; in the case of dynamic crypto map entries, if no SA exists, the packet is dropped.) After passing the regular access lists at the interface, inbound traffic is evaluated against the crypto access lists specified by the entries of the interface's crypto map set to determine if it should be protected by crypto and, if so, which crypto policy applies. (In the case of IPSec, unprotected traffic is discarded because it should have been protected by IPSec.)

In the case of IPSec, the access list is also used to identify the flow for which the IPSec security associations are established. In the outbound case, the **permit** entry is used as the data flow identity (in general), while in the inbound case the data flow identity specified by the peer must be "permitted" by the crypto access list.

## Examples

The following example shows the minimum required crypto map configuration when IKE will be used to establish the security associations. (This example is for a static crypto map.)

```
crypto map mymap 10 ipsec-isakmp
 match address 101
 set transform-set my_t_set1
 set peer 10.0.0.1
```

## Related Commands

| Command | Description |
|---|---|
| **crypto dynamic-map** | Creates a dynamic crypto map entry and enters the crypto map configuration command mode. |
| **crypto map (global IPSec)** | Creates or modifies a crypto map entry and enters the crypto map configuration mode. |
| **crypto map (interface IPSec)** | Applies a previously defined crypto map set to an interface. |
| **crypto map local-address** | Specifies and names an identifying interface to be used by the crypto map for IPSec traffic. |
| **set peer (IPSec)** | Specifies an IPSec peer in a crypto map entry. |
| **set pfs** | Specifies that IPSec should ask for perfect forward secrecy (PFS) when requesting new security associations for this crypto map entry, or that IPSec requires PFS when receiving requests for new security associations. |
| **set security-association level per-host** | Specifies that separate IPSec security associations should be requested for each source/destination host pair. |
| **set security-association lifetime** | Overrides (for a particular crypto map entry) the global lifetime value, which is used when negotiating IPSec security associations. |
| **set session-key** | Specifies the IPSec session keys within a crypto map entry. |
| **set transform-set** | Specifies which transform sets can be used with the crypto map entry. |
| **show crypto map (IPSec)** | Displays the crypto map configuration. |

# match authentication trustpoint

To specify the trustpoint name that should be used to authenticate the SDP peer's certificate, use the **match authentication trustpoint** command in tti-registrar configuration mode. To remove this configuration, use the **no** form of this command.

**match authentication trustpoint** *trustpoint-name*

**no match authentication trustpoint** *trustpoint-name*

## Syntax Description

| | |
|---|---|
| *trustpoint-name* | Specifies the truspoint name. |

## Command Default

No trustpoint name is specified for the iPhone deployment.

## Command Modes

Tti-registrar configuration mode (tti-registrar)

## Command History

| Release | Modification |
|---|---|
| 15.1(2)T | This command was introduced. |

## Usage Guidelines

The **match authentication trustpoint** command can be used optionally in the SDP registrar configuration, which is used to deploy Apple iPhones on a corporate network.

If the trustpoint name is not specified, then the trustpoint configured using the **authentication trustpoint** in tti-registrar configuration mode is used to authenticate the SDP peer's certificate.

## Examples

The following example configures the SDP registrar to run HTTPS in order to deploy Apple iPhones on a corporate network from global configuration mode:

```
Router(config)# crypto provisioning registrar
Router(tti-registrar)# url-profile start START
Router(tti-registrar)# url-profile intro INTRO
Router(tti-registrar)# match url /sdp/intro
Router(tti-registrar)# match authentication trustpoint apple-tp
Router(tti-registrar)# match certificate cat 10
Router(tti-registrar)# mime-type application/x-apple-aspen-config
Router(tti-registrar)# template location flash:intro.mobileconfig
Router(tti-registrar)# template variable p iphone-vpn
```

## Related Commands

| Command | Description |
|---|---|
| **crypto provisioning registrar** | Configures a device to become a registrar for the SDP exchange and enters tti-registrar configuration mode. |
| **url-profile** | Specifies a URL profile that configures the SDP registrar to run HTTPS in order to deploy Apple iPhones on a corporate network. |

| Command | Description |
|---|---|
| **match url** | Specifies the URL to be associated with the URL profile. |
| **authentication trustpoint** | Specifies the trustpoint used to authenticate the SDP petitioner device's existing certificate. |
| **match certificate** | Enters the name of the certificate map used to authorize the peer's certificate. |
| **mime-type** | Specifies the MIME type that the SDP registrar should use to respond to a request received through the URL profile. |
| **template location** | Specifies the location of the template that the SDP Registrar should use while responding to a request received through the URL profile. |
| **template variable p** | Specifies the value that goes into the OU field of the subject name in the certificate to be issued. |

# match body regex

To specify an arbitrary text expression to restrict specified content-types and content encoding types for text and HTML in the "body" of the e-mail, use the **match body regex** command in class-map configuration mode. To remove this match criterion, use the **no** form of this command.

**match body regex** *parameter-map-name*

**no match body regex** *parameter-map-name*

## Syntax Description

| | |
|---|---|
| *parameter-map-name* | Name of a specific traffic pattern specified through the **parameter-map type regex** command. |

## Command Default

None

## Command Modes

Class-map configuration

## Command History

| Release | Modification |
|---|---|
| 12.4(9)T | This command was introduced. |
| Cisco IOS XE Release 3.2S | This command was integrated into Cisco IOS XE Release 3.2S. |

## Usage Guidelines

If a match is found, possible actions that can be specified within the policy are as follows: allow, reset, or log. (The log action triggers a syslog message when a match is found.)

The text or HTML pattern is scanned only if the encoding is 7-bit or 8-bit and the encoding is checked before attempting to match the pattern. If the pattern is of another encoding type (e.g. base64, zip files etc.), then the pattern cannot be scanned

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

## Examples

The following example shows how to configure an SMTP policy to block an e-mail that contains the pattern "*UD-421590*" in the body of an e-mail.

```
parameter-map type regex doc-data
pattern "*UD-421590*"

class-map type inspect smtp c1
match body regex doc-data

policy-map type inspect smtp p1
class type inspect smtp c1
log
```

**Related Commands**

| Command | Description |
|---|---|
| **class-map type inspect smtp** | Creates a class map for the SMTP protocol so that the match criteria is set to match criteria for this class map. |
| **policy-map type inspect smtp** | Create a Layer 7 SMTP policy map. |

# match certificate

To specify the name of the certificate map used to authorize the peer's certificate, use the **match certificate** command in tti-registrar configuration mode. To remove this configuration, use the **no** form of this command.

**match certificate** *certificate-map*

**no match certificate** *certificate-map*

## Syntax Description

| | |
|---|---|
| *certificate-map* | Specifies the certificate map name. |

## Command Default

No certificate map name is specified for the iPhone deployment.

## Command Modes

Tti-registrar configuration mode (tti-registrar)

## Command History

| Release | Modification |
|---|---|
| 15.1(2)T | This command was introduced. |

## Usage Guidelines

The **match certificate** command can be used optionally in the SDP registrar configuration, which is used to deploy Apple iPhones on a corporate network.

## Examples

The following example configures the SDP registrar to run HTTPS in order to deploy Apple iPhones on a corporate network from global configuration mode:

```
Router(config)# crypto provisioning registrar
Router(tti-registrar)# url-profile start START
Router(tti-registrar)# url-profile intro INTRO
Router(tti-registrar)# match url /sdp/intro
Router(tti-registrar)# match authentication trustpoint apple-tp
Router(tti-registrar)# match certificate cat 10
Router(tti-registrar)# mime-type application/x-apple-aspen-config
Router(tti-registrar)# template location flash:intro.mobileconfig
Router(tti-registrar)# template variable p iphone-vpn
```

## Related Commands

| Command | Description |
|---|---|
| **crypto provisioning registrar** | Configures a device to become a registrar for the SDP exchange and enters tti-registrar configuration mode. |
| **url-profile** | Specifies a URL profile that configures the SDP registrar to run HTTPS in order to deploy Apple iPhones on a corporate network. |
| **match url** | Specifies the URL to be associated with the URL profile. |

| Command | Description |
|---|---|
| **match authentication trustpoint** | Specifies the trustpoint name that should be used to authenticate the SDP peer's certificate in order to deploy Apple iPhones on a corporate network. |
| **mime-type** | Specifies the MIME type that the SDP registrar should use to respond to a request received through the URL profile. |
| **template location** | Specifies the location of the template that the SDP Registrar should use while responding to a request received through the URL profile. |
| **template variable p** | Specifies the value that goes into the OU field of the subject name in the certificate to be issued. |

# match certificate (ca-trustpoint)

To associate a certificate-based access control list (ACL) that is defined with the **crypto ca certificate map** command, use the **match certificate** command in ca-trustpoint configuration mode. To remove the association, use the **no** form of this command.

**match certificate** *certificate-map-label* [**allow expired-certificate | skip revocation-check | skip authorization-check**]

**no match certificate** *certificate-map-label* [**allow expired-certificate | skip revocation-check | skip authorization-check**]

## Syntax Description

| | |
|---|---|
| *certificate-map-label* | Matches the *label* argument specified in a previously defined **crypto ca certificate map** command. |
| **allow expired-certificate** | (Optional) Ignores expired certificates.<br>**Note** If this keyword is not configured, the router does not ignore expired certificates. |
| **skip revocation-check** | (Optional) Allows a trustpoint to enforce certificate revocation lists (CRLs) except for specific certificates.<br>**Note** If this keyword is not configured, the trustpoint enforces CRLs for all certificates. |
| **skip authorization-check** | (Optional) Skips the authentication, authorization, and accounting (AAA) check of a certificate when public key infrastructure (PKI) integration with an AAA server is configured.<br>**Note** If this keyword is not configured and PKI integration with an AAA server is configured, the AAA checking of a certificate is done. |

## Defaults

If this command is not configured, no default match certificate is configured. Each of the **allow expired-certificate**, **skip revocation-check**, and **skip authorization-check** keywords have a default (see the "Syntax Description" section).

## Command Modes

Ca-trustpoint configuration

## Command History

| Release | Modification |
|---|---|
| 12.2(15)T | This command was introduced. |
| 12.2(18)SXD | This command was integrated into Cisco IOS Release 12.2(18)SXD. |
| 12.3(4)T | The **allow expired-certificate**, **skip revocation-check**, and **skip authorization-check** keywords were added. |
| 12.2(33)SRA | This command was integrated into Cisco IOS release 12.(33)SRA. |

## Usage Guidelines

The **match certificate** command associates the certificate-based ACL defined with the **crypto ca certificate map** command to the trustpoint. The *certificate-map-label* argument in the **match certificate** command must match the *label* argument specified in a previously defined **crypto ca certificate map** command.

The certificate map with the label *certificate-map-label* must be defined before it can be used with the **match certificate** subcommand.

A certificate referenced in a **match certificate** command may not be deleted until all references to the certificate map are removed from configured trustpoints (that is, no **match certificate** commands can reference the certificate map being deleted).

When the certificate of a peer has been verified, the certificate-based ACL as specified by the certificate map is checked. If the certificate of the peer matches the certificate ACL, or a certificate map is not associated with the trustpoint used to verify the certificate of the peer, the certificate of the peer is considered valid.

If the certificate map does not have any attributes defined, the certificate is rejected.

### Using the allow expired-certificate Keyword

The **allow expired-certificate** keyword has two purposes:

- If the certificate of a peer has expired, this keyword may be used to "allow" the expired certificate until the peer is able to obtain a new certificate.
- If your router clock has not yet been set to the correct time, the certificate of a peer will appear to be not yet valid until the clock is set. This keyword may be used to allow the certificate of the peer even though your router clock is not set.



**Note**

- If Network Time Protocol (NTP) is available only via the IPSec connection (usually via the hub in a hub-and-spoke configuration), the router clock can never be set. The tunnel to the hub cannot be "brought up" because the certificate of the hub is not yet valid.
- "Expired" is a generic term for a certificate that is expired or that is not yet valid. The certificate has a start and end time. An expired certificate, for purposes of the ACL, is one for which the current time of the router is outside the start and end time specified in the certificate.

### Using the skip revocation-check Keyword

The type of enforcement provided using the **skip revocation-check** keyword is most useful in a hub-and-spoke configuration in which you also want to allow direct spoke-to-spoke connections. In pure hub-and-spoke configurations, all spokes connect only to the hub, so CRL checking is necessary only on the hub. If one spoke communicates directly with another spoke, the CRLs must be checked. However, if the trustpoint is configured to require CRLs, the connection to the hub to retrieve the CRL usually cannot be made because the CRL is available only via the connection hub.

### Using the skip authorization-check Keyword

If the communication with an AAA server is protected with a certificate, and you want to skip the AAA check of the certificate, use the **skip authorization-check** keyword. For example, if a Virtual Private Network (VPN) tunnel is configured so that all AAA traffic goes over that tunnel, and the tunnel is protected with a certificate, you can use the **skip authorization-check** keyword to skip the certificate check so that the tunnel can be established.

The **skip authorization-check** keyword should be configured after PKI integration with an AAA server is configured.

## Examples

The following example shows a certificate-based ACL with the label "Group" defined in a **crypto ca certificate map** command and included in the **match certificate** command:

```
crypto ca certificate map Group 10
 subject-name co ou=WAN
 subject-name co o=Cisco
!
crypto ca trustpoint pki
 match certificate Group
```

The following example shows a configuration for a central site using the **allow expired-certificate** keyword. The router at a branch site has an expired certificate named "branch1" and has to establish a tunnel to the central site to renew its certificate.

```
crypto pki trustpoint VPN-GW
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Central VPN Gateway
 revocation-check crl
 match certificate branch1 allow expired-certificate
```

The following example shows a branch office configuration using the **skip revocation-check** keyword. The trustpoint is being allowed to enforce CRLs except for "central-site" certificates.

```
crypto pki trustpoint home-office
 enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
 serial-number none
 fqdn none
 ip-address none
 subject-name o=Home Office Inc,cn=Branch 1
 revocation-check crl
 match certificate central-site skip revocation-check
```

The following example shows a branch office configuration using the **skip authorization-check** keyword. The trustpoint is being allowed to skip AAA checking for the central site.

```
crypto pki trustpoint home-office
 auth list allow_list
 auth user subj commonname
 match certificate central-site skip authorization-check
```

## Related Commands

| Command | Description |
|---|---|
| **crypto ca certificate map** | Defines certificate-based ACLs. |
| **crypto ca trustpoint** | Declares the CA that your router should use. |

# match certificate (ISAKMP)

To assign an Internet Security Association Key Management Protocol (ISAKMP) profile to a peer on the basis of the contents of arbitrary fields in the certificate, use the **match certificate** command in crypto ISAKMP profile configuration mode. To remove the profile, use the **no** form of this command.

**match certificate** *certificate-map*

**no match certificate** *certificate-map*

## Syntax Description

| | |
|---|---|
| *certificate-map* | Name of the certificate map. |

## Defaults

No default behavior or values

## Command Modes

Crypto ISAKMP profile configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(8)T | This command was introduced. |
| 12.2(33)SRA | This command was integrated into Cisco IOS Release 12.2(33)SRA. |
| 12.2(33)SX | This command is supported in the Cisco 12.2SX family of releases. Support in a 12.2SX release is dependent on your feature set, platform, and platform hardware. |

## Usage Guidelines

The **match certificate** command is used after the certificate map has been configured and the ISAKMP profiles have been assigned to them.

## Examples

The following configuration example shows that whenever a certificate contains “ou = green,” the ISAKMP profile “cert_pro” will be assigned to the peer.

```
crypto pki certificate map cert_map 10
 subject-name co ou = green
!
!
crypto isakmp identity dn
crypto isakmp profile cert_pro
   ca trust-point 2315
   ca trust-point LaBcA
   initiate mode aggressive
   match certificate cert_map
```

## Related Commands

| Command | Description |
|---|---|
| **client configuration group** | Associates a group with the peer that has been assigned an ISAKMP profile. |

# match certificate override cdp

To manually override the existing certificate distribution point (CDP) entries for a certificate with a URL or directory specification, use the **match certificate override cdp** command in ca-trustpoint configuration mode. To remove the override, use the **no** form of this command.

**match certificate** *certificate-map-label* **override cdp** {**url** | **directory**} *string*

**no match certificate** *certificate-map-label* **override cdp** {**url** | **directory**} *string*

## Syntax Description

| | |
|---|---|
| *certificate-map-label* | A user-specified label that must match the *label* argument specified in a previously defined **crypto ca certificate map** command. |
| **url** | Specifies that the certificates CDPs will be overridden with an http or ldap URL. |
| **directory** | Specifies that the certificate's CDPs will be overridden with an ldap directory specification. |
| *string* | The URL or directory specification. |

## Defaults

The existing CDP entries for the certificate are used.

## Command Modes

Ca-trustpoint configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(7)T | This command was introduced. |
| 12.2(18)SXE | This command was integrated into Cisco IOS Release 12.2(18)SXE. |
| 12.2(33)SRA | This command was integrated into Cisco IOS release 12.(33)SRA. |

## Usage Guidelines

Use the **match certificate override cdp** command to replace all of the existing CDPs in a certificate with a manually configured CDP URL or directory specification.

The *certificate-map-label* argument in the **match certificate override cdp** command must match the *label* argument specified in a previously defined **crypto ca certificate map** command.

**Note** Some applications may time out before all CDPs have been tried and will report an error message. This will not affect the router, and the Cisco IOS software will continue attempting to retrieve a CRL until all CDPs have been tried.

## Examples

The following example uses the **match certificate override cdp** command to override the CDPs for the certificate map named Group1 defined in a **crypto ca certificate map** command:

```
crypto ca certificate map Group1 10
 subject-name co ou=WAN
 subject-name co o=Cisco
!
crypto ca trustpoint pki
 match certificate Group1 override cdp url http://server.cisco.com
```

## Related Commands

| Command | Description |
|---|---|
| **crypto ca certificate map** | Defines certificate-based ACLs. |
| **crypto ca trustpoint** | Declares the CA that your router should use. |

# match certificate override ocsp

To override an Online Certificate Status Protocol (OCSP) server setting specified in either the Authority Info Access (AIA) field of the client certificate or in the trustpoint configuration, use the **match certificate override ocsp** command in ca-trustpoint configuration mode. To remove the OCSP server override setting, use the **no** form of this command.

**match certificate** *certificate-map-label* **override ocsp** [**trustpoint** *trustpoint-label*] *sequence-number* **url** *ocsp-url*

**no match certificate** *certificate-map-label* **override ocsp** [**trustpoint** *trustpoint-label*] *sequence-number* **url** *ocsp-url*

## Syntax Description

| | |
|---|---|
| *certificate-map-label* | Specifies the exact name of an existing certificate map label. |
| **trustpoint** *trustpoint-label* | (Optional) Specifies the existing trustpoint to be used when validating the OCSP server responder certificate. |
| *sequence-number* | Indicates the order of the override statements to be applied when a certificate is being verified.<br>**Note** Certificate matches are performed from the lowest sequence number to the highest sequence number. If more than one command is issued with the same sequence number, the previous OCSP server override setting is replaced. |
| **url** *ocsp-url* | Specifies the OCSP server URL. |

## Command Default

No override OSCP server setting will be configured.

## Command Modes

Ca-trustpoint configuration (ca-trustpoint)

## Command History

| Release | Modification |
|---|---|
| 12.4(6)T | This command was introduced. |
| Cisco IOS XE Release 2.4 | This command was implemented on the Cisco ASR 1000 series routers. |

## Usage Guidelines

OCSP server validation is usually based on the root certification authority (CA) certificate or a valid subordinate CA certificate, but may also be configured for validation of the OCSP server identity with the **match certificate override ocsp** command and **trustpoint** keyword.

One or more OCSP servers may be specified, either per client certificate or per group of client certificates. When the certificate matches a configured certificate map, the AIA field of the client certificate and any previously issued **ocsp url** command settings are overwritten with the specified OCSP server. If the **ocsp url** configuration exists and no map-based match occurs, the **ocsp url** configuration settings will continue to apply to the client certificates.

## Examples

The following example shows an excerpt of the running configuration output when adding an override OCSP server to the beginning of an existing sequence:

```
match certificate map3 override ocsp 5 url http://192.168.2.3/
show running-config
.
.
.
        match certificate map3 override ocsp 5 url http://192.168.2.3/
        match certificate map1 override ocsp 10 url http://192.168.2.1/
        match certificate map2 override ocsp 15 url http://192.168.2.2/
```

The following example shows an excerpt of the running configuration output when an existing override OSCP server is replaced and a trustpoint is specified to use an alternative public key infrastructure (PKI) hierarchy:

```
match certificate map4 override ocsp trustpoint tp4 10 url http://192.168.2.4/newvalue\
show running-config
.
.
.
        match certificate map3 override ocsp trustpoint tp3 5 url http://192.168.2.3/
        match certificate map1 override ocsp trustpoint tp1 10 url http://192.168.2.1/
        match certificate map4 override ocsp trustpoint tp4 10 url
http://192.168.2.4/newvalue
        match certificate map2 override ocsp trustpoint tp2 15 url http://192.168.2.2/
```

The following example shows an excerpt of the running configuration output when an existing override OCSP server is removed from an existing sequence:

```
no match certificate map1 override ocsp trustpoint tp1 10 url http://192.168.2.1/
show running-config
.
.
.
        match certificate map3 override ocsp trustpoint tp3 5 url http://192.168.2.3/
        match certificate map4 override ocsp trustpoint tp4 10 url
http://192.168.2.4/newvalue
        match certificate map2 override ocsp trustpoint tp2 15 url http://192.168.2.2/
```

## Related Commands

| Command | Description |
|---|---|
| **crypto pki certificate map** | Defines values in a certificate that should be matched or not matched. |
| **ocsp url** | Specifies the URL of an OCSP server so that the trustpoint can check the certificate status. |

# match certificate override sia

To manually override the existing SubjectInfoAccess (SIA) attribute, use the **match certificate override sia** command in CA-trustpoint configuration mode. To remove the override, use the **no** form of this command.

**match certificate** *certificate-map-label* **override sia** *sequence-number certificate-url*

**no match certificate** *certificate-map-label* **override sia**

## Syntax Description

| | |
|---|---|
| *certificate-map-label* | A user-specified label that should match the label argument specified in a previously defined **crypto ca certificate map** command. |
| *sequence-number* | The order of the override statements to be applied when a certificate is being verified.<br>**Note** Certificate matches are performed from the lowest sequence number to the highest sequence number. If more than one command is issued with the same sequence number, the previous SIA override setting is replaced. |
| *certificate-url* | The remote location of the certificate in URL format. |

## Command Default

The existing SIA entries for the certificate are used.

## Command Modes

CA-trustpoint configuration (ca-trustpoint)

## Command History

| Release | Modification |
|---|---|
| 15.1(2)T | This command was introduced. |

## Usage Guidelines

The certificate's storage location is contained in the certificate itself by the issuing authority. This data is contained in the SIA and the AuthorityInfoAccess (AIA) extension in certificates. Use the **match certificate override sia** command to manually configure the remote location of the identity certificate regardless of the SIA attribute in the certificate.

## Examples

The following example shows how to use the **match certificate override sia** command to override the SIAs for the certificate map named Group1 defined in a **crypto ca certificate map** command:

```
Router(config)# crypto ca certificate map Group1 10
Router(ca-certificate-map)# subject-name co ou=WAN
Router(ca-certificate-map)# subject-name co o=Cisco
!
Router(config)# crypto ca trustpoint pki
Router (ca-trustpoint)# match certificate Group1 override sia 100
http://certs.example.com/certificate.cer
```

**Related Commands**

| Command | Description |
|---|---|
| **crypto ca certificate map** | Defines certificate-based ACLs. |
| **crypto ca trustpoint** | Declares the CA that your router should use. |

# match class-map

To use a traffic class as a classification policy, use the **match class-map** command in class-map configuration mode. To remove a specific traffic class as a match criterion, use the **no** form of this command.

**match class-map** *class-map-name*

**no match class-map** *class-map-name*

## Syntax Description

| | |
|---|---|
| *class-map-name* | Name of the traffic class to use as a match criterion. |

## Command Default

No match criteria are specified.

## Command Modes

Class-map configuration

## Command History

| Release | Modification |
|---|---|
| 12.0(5)XE | This command was introduced. |
| 12.1(1)E | This command was integrated into Cisco IOS Release 12.1(1)E. |
| 12.1(5)T | This command was integrated into Cisco IOS Release 12.1(5)T. |
| 12.4(6)T | This command was enhanced to support Zone-Based Policy Firewall. |
| 12.2(33)SRA | This command was integrated into Cisco IOS Release 12.2(33)SRA. |
| 12.2(31)SB | This command was implemented on the Cisco 10000 series. |
| 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. |
| Cisco IOS XE Release 3.2S | This command was integrated into Cisco IOS XE Release 3.2S. |

## Usage Guidelines

The only method of including both match-any and match-all characteristics in a single traffic class is to use the **match class-map** command. To combine match-any and match-all characteristics into a single class, do one of the following:

- Create a traffic class with the match-any instruction and use a class configured with the match-all instruction as a match criterion (using the **match class-map** command).
- Create a traffic class with the match-all instruction and use a class configured with the match-any instruction as a match criterion (using the **match class-map** command).

You can also use the **match class-map** command to nest traffic classes within one another, saving users the overhead of re-creating a new traffic class when most of the information exists in a previously configured traffic class.

When packets are matched to a class map, a traffic rate is generated for these packets. In a zone-based firewall policy, only the first packet that creates a session matches the policy. Subsequent packets in this flow do not match the filters in the configured policy, but instead match the session directly. The statistics related to subsequent packets are shown as part of the 'inspect' action.

## Examples

### Non-Zone-Based Policy Firewall Examples

In the following example, the traffic class called class1 has the same characteristics as traffic class called class2, with the exception that traffic class class1 has added a destination address as a match criterion. Rather than configuring traffic class class1 line by line, you can enter the **match class-map class2** command. This command allows all of the characteristics in the traffic class called class2 to be included in the traffic class called class1, and you can simply add the new destination address match criterion without reconfiguring the entire traffic class.

```
Router(config)# class-map match-any class2
Router(config-cmap)# match protocol ip
Router(config-cmap)# match qos-group 3
Router(config-cmap)# match access-group 2
Router(config-cmap)# exit

Router(config)# class-map match-all class1
Router(config-cmap)# match class-map class2
Router(config-cmap)# match destination-address mac 1.1.1
Router(config-cmap)# exit
```

The following example shows how to combine the characteristics of two traffic classes, one with match-any and one with match-all characteristics, into one traffic class with the **match class-map** command. The result of traffic class called class4 requires a packet to match one of the following three match criteria to be considered a member of traffic class called class 4: IP protocol *and* QoS group 4, destination MAC address 1.1.1, or access group 2. Match criteria IP protocol *and* QoS group 4 are required in the definition of the traffic class named class3 and included as a possible match in the definition of the traffic class named class4 with the **match class-map class3** command.

In this example, only the traffic class called class4 is used with the service policy called policy1.

```
Router(config)# class-map match-all class3
Router(config-cmap)# match protocol ip
Router(config-cmap)# match qos-group 4
Router(config-cmap)# exit

Router(config)# class-map match-any class4
Router(config-cmap)# match class-map class3
Router(config-cmap)# match destination-address mac 1.1.1
Router(config-cmap)# match access-group 2
Router(config-cmap)# exit

Router(config)# policy-map policy1
Router(config-pmap)# class class4
Router(config-pmap-c)# police 8100 1500 2504 conform-action transmit exceed-action
set-qos-transmit 4
Router(config-pmap-c)# exit
```

## Related Commands

| Command | Description |
|---|---|
| **class-map** | Creates a class map to be used for matching packets to a specified class. |

# match class session

To configure match criteria for a class map used to identify a session (flow) containing packets of interest, which is then applied to all packets transmitted during the session, use the **match class session** command in class map configuration mode. To remove this configuration, use the **no** form of this command.

**match class** *class-name* [**packet-range** *low high* | **byte-range** *low high*] **session**

**no match class** *class-name* [**packet-range** *low high* | **byte-range** *low high*] **session**

## Syntax Description

| | |
|---|---|
| *class-name* | Specifies the class map used to identify a session containing packets of interest. The classification results are preserved for the subsequent packets of the same packet session. |
| **packet-range** *low high* | (Optional) Specifies the range of packets from 1 to 2147483647, in which the regular expressions (regex) within every packet is are checked. The classification results are preserved for the specified packets or bytes of the same packet session. |
| **byte-range** *low high* | (Optional) Specifies the range of bytes from 1 to 2147483647, in which the regular expressions (regex) within every packet are checked. The classification results are preserved for the specified packets or bytes of the same packet session. |

## Command Default

The regex matching is within a single packet with a range 1 to infinity.

## Command Modes

Class map configuration (config-cmap)

## Command History

| Release | Modification |
|---|---|
| 15.1(3)T | This command was introduced. |

## Usage Guidelines

With the introduction of Cisco IOS Release15.1(3)T, Flexible Packet Matching (FPM) can now match every packet against the filters specified in the class map and pass the match result to consecutive packets of the same network session. If a filter matches with malicious content in the packet's protocol header or payload, then the required action is taken to resolve the problem.

The **match class session** command configures match criteria that identify a session containing packets of interest, which is then applied to all packets transmitted during the session. The **packet-range** and **byte-range** keywords are used to create a filter mechanism that increases the performance and matching accuracy of regex-based FPM class maps by classifying traffic that resides in the narrow packet number or byte ranges of each packet flow. If packets go beyond the classification window, then the packet flow can be identified as unknown and packet classification is terminated early to increase performance. For example, a specific application can be blocked efficiently by filtering all packets that belong to this application on a session. These packets are dropped without matching every individual packet with the filters, which improves the performance of a session.

These filters also reduce the number of false positives introduced by general regex-based approaches. For example, Internet company messenger traffic can be classified with a string like **intco**, **intcomsg**, and **ic**. These strings are searched for in a packet's payload. These small strings can appear in the packet payload of any other applications, such as e-mail, and can introduce false positives. False positives can be avoided by specifying which regex is searched within which packet of a particular packet flow.

Once the match criteria are applied to packets belonging to the specific traffic class, these packets can be discarded by configuring the **drop all** command in a policy map. Packets match only on the packet flow entry of an FPM, and skip user-configured classification filters.

A match class does not have to be applied exclusively for a regex-based filter. Any FPM filter can be used in the nested match class filter. For example, if the match class **c1** has the filter **match field TCP source-port eq 80**, then the **match class c1 session** command takes the same action for the packets that follow the first matching packet.

## Examples

The following example shows how to configure a class map and policy map to specify the protocol stack class, the match criteria and action to take, and a combination of classes using session-based (flow-based) and nonsession-based actions. The **drop all** command is associated with the action to be taken on the policy.

```
Router(config)# class-map type access-control match-all my-HTTP
Router(config-cm)# match field tcp destport eq 8080
Router(config-cm)# match start tcp payload-start offset 20 size 10 regex "GET"

Router(config)# class-map type access-control match-all my-FTP
Router(config-cmap)# match field tcp destport eq 21

Router(config)# class-map type access-control match all class1
Router(config-cmap)# match class my-HTTP session
Router(config-cmap)# match start tcp payload-start offset 40 size 20 regex "abc.*def"

Router(config)# policy-map type access-control my_http_policy
Router(config-pmap)# class class1
Router(config-pmap-c)# drop all

Router(config)# interface gigabitEthernet 0/1
Router(config-if)# service-policy type access-control input my_http_policy
```

The following example shows how to configure a class map and policy map to specify the protocol stack class, the match criteria and action to take, and a combination of classes using session-based (flow-based) and nonsession-based actions. However, this example uses the **match class** command with the **packet-range** keyword, which acts as a filter mechanism to increases the performance and matching accuracy of the regex-based FPM class map.

```
Router(config)# load disk2:ip.phdf
Router(config)# load protocol disk2:tcp.phdf

Router(config)# class-map type stack match-all ip_tcp
Router(config-cmap)# description "match TCP over IP packets"
Router(config-cmap)# match field ip protocol eq 6 next tcp

Router(config)# class-map type access-control match-all WM
Router(config-cmap) # match start tcp payload-start offset 20 size 20 regex
".*(WEBCO|WMSG|WPNS).......[LWT].*\xc0\x80"

Router(config)# class-map type access-control match-all wtube
Router(config-cmap) # match start tcp payload-start offset 20 size 20 regex
".*GET\x20.*HTTP\x2f(0\.9|1\.0|1\.1)\x0d\x0aHost:\x20webtube.com\x0d\x0a"
```

```
Router(config)# class-map type access-control match-all doom
Router(config-cmap) # match start tcp payload-start offset 20 size 20 string virus

Router(config)# class-map type access-control match-all class_webco
Router(config-cmap)# match class WM session
Router(config-cmap)# match field ip length eq 0x194
Router(config-cmap)# match start network-start offset 224 size 4 eq 0x4011010

Router(config)# class-map type access-control match-all class_webtube
Router(config-cmap)# match class wtube packet-range 1 5 session
Router(config-cmap)# match class doom session
Router(config-cmap)# match field ip length eq 0x194
Router(config-cmap)# match start network-start offset 224 size 4 eq 0x4011010

Router(config)# policy-map type access-control my_policy
Router(config-pmap)# class class_webco
Router(config-pmap-c)# log

Router(config)# policy-map type access-control my_policy
Router(config-pmap)# class class_webtube
Router(config-pmap-c)# drop all

Router(config)# policy-map type access-control P1
Router(config-pmap)# class ip_tcp
Router(config-pmap-c)# service-policy my_policy

Router(config)# interface gigabitEthernet 0/1
Router(config-if)# service-policy type access-control input P1
```

## Related Commands

| Command | Description |
|---|---|
| **drop** | Configures a traffic class to discard packets belonging to a specific class. |
| **log** | Generates log messages for the traffic class. |

# match cmd

To specify a value that limits the length of the ESMTP command line or specifies the ESMTP command line verb used to thwart denial of service (DoS) attacks, use the **match cmd** command in class-map configuration mode. To disable this inspection parameter, use the **no** form of this command.

**match cmd** {**line length gt** *length* | **verb** {**AUTH** | **DATA** | **EHLO** | **ETRN** | **EXPN** | **HELO** | **HELP** | **MAIL NOOP** | **QUIT** | **RCPT** | **RSET** | **SAML** | **SEND** | **SOML** | **STARTTLS** | **VERB** | **VRFY** | **WORD**}}

**no match cmd** {**line length gt** *length* | **verb** {**AUTH** | **DATA** | **EHLO** | **ETRN** | **EXPN** | **HELO** | **HELP** | **MAIL NOOP** | **QUIT** | **RCPT** | **RSET** | **SAML** | **SEND** | **SOML** | **STARTTLS** | **VERB** | **VRFY** | **WORD**}}

Syntax Description

| | |
|---|---|
| **line length gt** *length* | Specifies the ESMTP command line greater than the length of a number of characters from 1 to 65535. |
| **verb** | Specifies the ESMTP command verb used to thwart DoS attacks. |
| **AUTH** | SMTP service extension whereby an SMTP client may indicate an authentication mechanism to the server, perform an authentication protocol exchange, and optionally negotiate a security layer for subsequent protocol interactions. |
| **DATA** | Sent by a client to initiate the transfer of message content. |
| **EHLO** | Enables the server to identify its support for Extended Simple Mail Transfer Protocol (ESMTP) commands. |
| **ETRN** | Requests the local SMTP server to initiate delivery of mail to the external SMTP server on a separate SMTP connection. |
| **EXPN** | Expand a mailing list address into individual recipients. Often disabled to prevent use by spammers. |
| **HELO** | Sent by a client to identify itself, usually with a domain name. |
| **HELP** | Returns a list of commands that are supported by the SMTP service. |
| **MAIL NOOP** | Start of MAIL FROM: Identifies sender of mail message. May be forged. May not correspond to the From: line in a mail message. Should be added in Return Path header. Address to send any undeliverable notifications (bounces). The NO OPeration (NOOP) does nothing, except keep the connection active and help synchronize commands and responses. |
| **QUIT** | Terminates the session. |
| **RCPT** | Identifies the message recipients; used in the form RCPT TO: |
| **RSET** | Nullifies the entire message transaction and resets the buffer. |
| **SAML** | Start of SAML FROM: Like MAIL except supposed to also display the message on the recipients computer (early form of instant messaging). |
| **SOML** | Start of SAML FROM: Like MAIL except supposed to either mail the message OR display the message on the recipients computer (early form of instant messaging) |
| **STARTTLS** | Triggers start of TLS negotiation for secure SMTP conversation. If successful, resets state to before EHLO command sent. |
| **VERB** | Enables verbose (detailed) responses. |

| | |
|---|---|
| **VRFY** | Verifies that a mailbox is available for message delivery; for example, the **VRFY MARK** command verifies that a mailbox for MARK resides on the local server. This command is off by default in Exchange implementations. |
| **WORD** | Specifies a word in the body of the e-mail message. |

## Command Default

The length of the ESMTP command line or command line verb is not defined.

## Command Modes

Class-map configuration

## Command History

| Release | Modification |
|---|---|
| 12.4(20)T | This command was introduced. |
| Cisco IOS XE Release 3.2S | This command was integrated into Cisco IOS XE Release 3.2S. |

## Usage Guidelines

In a **class-map type inspect smtp match-all** command statement with the **match cmd verb** command statement, only the following **match cmd line length gt** command statement can coexist. For example:

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

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

The class-map **c2** matches if the length of only the e-mail command is greater than 256 bytes (which is not applicable to other commands), which translates to: If the length of the MAIL command exceeds the configured value.

**Note** If no **match cmd verb** command statement is specified in a **class-map type inspect smtp match-all** command statement for a class-map, which contains the **match cmd line length gt** command statement, then the class-map applies to all SMTP commands.

## Examples

The following example shows how to configure an SMTP application firewall policy to limit the length of an SMTP command line to prevent a Denial of Service (DoS) attack:

```
class-map type inspect smtp c1
 match header length gt 16000
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect smtp** | Creates a class map for the SMTP protocol so that the match criteria is set to match criteria for this class map. |

# match data-length

To determine if the amount of data transferred in a Simple Mail Transfer Protocol (SMTP) connection is greater than the configured limit, use the **match data-length** command in class-map type inspect smtp configuration mode. To remove this match criteria, use the **no** form of this command.

**match data-length gt** *max-data-value*

**no match data-length gt** *max-data-value*

## Syntax Description

| | |
|---|---|
| **gt** *max-data-value* | Maximum number of bytes (data) that can be transferred in a single SMTP session. After the maximum value is exceeded, the firewall logs an alert message and closes the session. The default is 20. |

## Command Default

The inspection rule is not defined.

## Command Modes

Class-map type inspect smtp configuration

## Command History

| Release | Modification |
|---|---|
| 12.4(6)T | This command was introduced. |
| Cisco IOS XE Release 3.2S | This command was integrated into Cisco IOS XE Release 3.2S. |

## Usage Guidelines

The **match data-length** match criteria can be specified only under an SMTP class map. For more information, see the **class-map type inspect smtp** command.

## Examples

The following example specifies that a maximum of 200000 bytes can be transferred in a single SMTP session:

```
class-map type inspect smtp c11
 match data-length gt 200000

policy-map type inspect smtp p11
 class type inspect smtp c11
  reset
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect smtp** | Configures inspection parameters for SMTP. |
| **ip inspect name** | Defines a set of inspection rules. |

# match encrypted

To configure the match criteria for a class map on the basis of encrypted Flexible Packet Matching (FPM) filters and enter FPM match encryption filter configuration mode, use the **match encrypted** command in class-map configuration mode. To remove the specified match criteria, use the **no** form of this command.

**match encrypted**

**no match encrypted**

## Syntax Description

This command has no arguments or keywords.

## Command Default

No match criteria are configured.

## Command Modes

Class-map configuration (config-cmap)

## Command History

| Release | Modification |
|---|---|
| 15.0(1)M | This command was introduced. |

## Usage Guidelines

If you have access to an encrypted traffic classification definition file (eTCDF) or if you know valid values to configure encrypted Flexible Packet Matching (FPM) filters, you can configure the same eTCDF through the command-line interface instead of using the preferred method of loading the eTCDF on the router. You must create a class map of type access-control using the **class-map type** command, and use the **match encrypted** command to configure the match criteria for the class map on the basis of encrypted FPM filters and enter FPM match encryption filter configuration mode. You can then use the appropriate commands to specify the algorithm, cipher key, cipher value, filter hash, filter ID, and filter version. You can copy the values from the eTCDF by opening the eTCDF in any text editor.

## Examples

The following example shows how to enter FPM match encryption filter configuration mode:

```
Router(config)# class-map type access-control match-all class2
Routre(config-cmap)# match encrypted
Router(c-map-match-enc-config)#
```

## Related Commands

| Command | Description |
|---|---|
| **algorithm** | Specifies the algorithm to be used for decrypting the filters. |
| **cipherkey** | Specifies the symmetric keyname that is used to decrypt the filter. |
| **ciphervalue** | Specifies the encrypted filter contents. |
| **class-map type** | Creates a class map to be used for matching packets to a specified class. |
| **filter-hash** | Specifies the hash for verification and validation of decrypted contents. |

| Command | Description |
|---|---|
| **filter-id** | Specifies a filter level ID for encrypted filters. |
| **filter-version** | Specifies the filter level version value for encrypted filters. |

# match file-transfer

To use file transfers as the match criterion, use the **match file-transfer** command in class-map configuration mode. To remove the file transfer match criterion from the configuration file, use the **no** form of this command.

**match file-transfer** [*regular-expression*]

**no match file-transfer** [*regular-expression*]

## Syntax Description

| | |
|---|---|
| *regular-expression* | (Optional) The regular expression used to identify file transfers for a specified P2P application. For example, entering ".exe" as the regular expression would classify the Gnutella file transfer connections containing the string ".exe" as matches for the traffic policy.<br><br>To specify that all file transfer connections be identified by the traffic class, use an asterisk (*) as the regular expression. |

## Command Default

None

## Command Modes

Class-map configuration

## Command History

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

## Usage Guidelines

After the **class-map type inspect** command is issued and a P2P application is specified, you can use the **match file-transfer** command to configure the Cisco IOS Firewall to match file transfer connections within any supported P2P protocol.

**Note** This command can be used only with the following supported P2P protocols: eDonkey, Gnutella, Kazaa Version 2, and FastTrack.

## Examples

The following example shows how to configure the Cisco IOS Firewall to block and reset all Gnutella file transfers that are classified into the "my-gnutella-restrictions" class map:

```
class-map type inspect gnutella match-any my-gnutella-restrictions
 match file-transfer *
!
policy-map type inspect p2p my-p2p-policy
 reset
 log
```

**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. |

# match header count

To configure an HTTP firewall policy to permit or deny HTTP traffic on the basis of request, response, or both request and response messages whose headers do not exceed a maximum number of fields, use the **match header count** command in class-map configuration mode. To change the configuration, use the **no** form of this command.

**match** {**request** | **response** | **req-resp**} **header** [*header-name*] **count gt** *number*

**no match** {**request** | **response** | **req-resp**} **header** [*header-name*] **count gt** *number*

## Syntax Description

| | |
|---|---|
| **request** | Headers in request messages are checked for the match criterion. |
| **response** | Headers in response messages are checked for the match criterion. |
| **req-resp** | Headers in both request and response messages are checked for the match criterion. |
| *header-name* | (Optional) Specific line in the header field. This argument enables the firewall to scan for repeated header fields.<br>**Note** If this option is defined, the **gt** *number* option must be set to 1. |
| **gt** *number* | Message cannot be greater than the specified number of header lines (fields). |

## Command Default

HTTP header-lines are not considered when permitting or denying HTTP traffic.

## Command Modes

Class-map configuration

## Command History

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

## Usage Guidelines

Use the **match header count** command to configure an HTTP firewall policy match criterion on the basis of a maximum allowed header fields count.

If a match is found, possible actions that can be specified within the policy are as follows: allow, reset, or log. (The log action triggers a syslog message when a match is found.)

### Header Field Repetition Inspection

To enable the firewall policy to checks whether a request or response message has repeated header fields, use the *header-name* argument. This functionality can be used to prevent session smuggling.

## Examples

The following example shows how to configure an HTTP application firewall policy to block all requests that exceed 16 header fields:

```
class-map type inspect http hdr_cnt_cm
 match req-resp header count gt 16

policy-map type inspect http hdr_cnt_pm
 class type inspect http hdr_cnt_cm
 reset
```

The following example shows how to configure an HTTP application firewall policy to block a request or response that has multiple content-length header lines:

```
class-map type inspect http multi_occrns_cm
 match req-resp header content-length count gt 1

policy-map type inspect http multi_occrns_pm
 class type inspect http multi_occrns_cm
 reset
```

# match header length gt

To thwart DoS attacks, use the **match header length gt** command in class-map configuration mode. To disable this inspection parameter, use the **no** form of this command.

**match header length gt** *bytes*

**no match length gt** *bytes*

Syntax Description

| | |
|---|---|
| *bytes* | Specifies a value from 1 to 65535 that limits the maximum length of the SMTP header in bytes. |

**Command Default**

Header length is not considered when permitting or denying SMTP messages.

**Command Modes**

Class-map configuration

**Command History**

| Release | Modification |
|---|---|
| 12.4(6)T | This command was introduced. |
| 12.4(9)T | The *header-name* argument and the **req-resp** keyword were added. |
| 12.4(20)T | The **request**, **response**, and **req-resp** keywords were removed and the *header-name* argument was removed. This command now applies to SMTP only. |
| Cisco IOS XE Release 3.2S | This command was integrated into Cisco IOS XE Release 3.2S. |

**Usage Guidelines**

The **match header length** command matches on the maximum length of an SMTP header. If that number is exceeded, the match succeeds.

If a match is found, possible actions that can be specified within the policy are as follows: allow, reset, or log. (The log action triggers a syslog message when a match is found.)

**Examples**

The following example shows how to configure an SMTP application firewall policy to block all SMTP headers that exceed a length of 4096 bytes:

```
class-map type inspect smtp c1
 match header length gt 4096

policy-map type inspect smtp p1
 class type inspect smtp c1
 reset
```

**Related Commands**

| Command | Description |
|---|---|
| **max-header-regex** | Specifies an arbitrary text expression in the SMTP e-mail message header (subject field) or e-mail body such as ‘subject’, ‘Received’, ‘To’ or other private header fields to monitor text patterns. |

# match header regex

To specify an arbitrary text expression (regular expression) in message or content type headers to monitor text patterns, use the **match header regex** command in class map configuration mode. To remove this filter from the configuration, use the **no** form of this command.

**Note** The **request**, **response**, and **req-resp** keywords and *header-name* argument are not used in the configuration of an SMTP class map.

**match** {**request | response | req-resp**} **header** [*header-name*] **regex** *parameter-map-name*

**no match** {**request | response | req-resp**} **header** [*header-name*] **regex** *parameter-map-name*

## Syntax Description

| | |
|---|---|
| **request** | Headers in request messages are checked for the match criterion. |
| **response** | Headers in response messages are checked for the match criterion. |
| **req-resp** | Headers in both request and response messages are checked for the match criterion. |
| *header-name* | Specific line or content type in the header field. This argument enables the firewall to scan for repeated header fields. |
| *parameter-map-name* | Name of a specific traffic pattern specified through the **parameter-map type regex** command. |

## Command Default

Policies do not monitor content type headers.

## Command Modes

Class-map configuration

## Command History

| Release | Modification |
|---|---|
| 12.4(9)T | This command was introduced. |
| 12.4(20)T | The **request**, **response**, and **req-resp** keywords and *header-name* argument were removed for the configuration of an SMTP class map. |
| Cisco IOS XE Release 3.2S | This command was integrated into Cisco IOS XE Release 3.2S. |

## Usage Guidelines

### Configuring a Class Map for SMTP

Use the **match header regex** command to configure an SMTP policy match criterion on the basis of headers that match the regular expression defined in a parameter map. An arbitrary text expression in the SMTP e-mail message header (subject field) or e-mail body such as ‘subject’, ‘Received’, ‘To’ or other private header fields helps the router to monitor text patterns.

### Configuring a Class Map for HTTP

An HTTP firewall policy match criteria can be configured on the basis of headers that match the regular expression defined in a parameter map.

HTTP has two regular expression (regex) options. One combines the **header** keyword, content type header name, and **regex** keyword and *parameter-map-name* argument. The other combines the **header** keyword and **regex** keyword and *parameter-map-name* argument.

- If the **header** and **regex** keywords are used with the *parameter-map-name* argument, it does not require a period and asterisk infront of the *parameter-map-name* argument. For example, either "html" or ".*html" *parameter-map-name* argument can be configured.
- If the **header** keyword is used with the **content-type** header name and **regex** keyword, then the parameter map name requires a period and asterisk (.*) in front of the *parameter-map-name* argument. For example, the *parameter-map-name* argument “html” is expressed as: .*html

**Note** If the period and asterisk is added in front of html (.*html), the *parameter-map-name* argument works for both HTTP regex options.

- The **mismatch** keyword is only valid for the **match response header content-type regex** command syntax for messages that need to be matched that have a **content-type** header name mismatch.

**Tip** It is a good practice to add ".*" to the **regex** *parameter-map-name* arguments that are not present at the beginning of a text string.

## Examples

### SMTP Class Map Example

The following example shows how to configure an SMTP policy using the **match header regex** command:

```
parameter-map type regex lottery-spam
 pattern "Subject:*lottery*"

class-map type inspect smtp c1
 match header regex lottery-spam

policy-map type inspect smtp p1
 class type inspect smtp c1
 reset
```

### HTTP Class Map Example

The following example shows how to configure an HTTP policy using the **match header regex** command:

```
parameter-map type inspect .*html

class-map type inspect http http-class
 match req-resp header regex .*html

policy-map type inspect http myhttp-policy
 class-type inspect http http-class
 reset
```

## Related Commands

| Command | Description |
|---|---|
| **max-header-regex** | Specifies an arbitrary text expression in the SMTP e-mail message header (subject field) or e-mail body such as ‘subject’, ‘Received’, ‘To’ or other private header fields to monitor text patterns. |
| **parameter-map type** | Creates or modifies a parameter map. |
| **policy-map type inspect** | Creates a Layer 3 and Layer 4 or a Layer 7 (protocol-specific) inspect type policy map. |

# match identity

To match an identity from a peer in an Internet Security Association and Key Management Protocol (ISAKMP) profile, use the **match identity** command in ISAKMP profile configuration mode. To remove the identity, use the **no** form of this command.

**match identity** {**group** *group-name* | **address** {*address* [*mask*] [*fvrf*] | **ipv6** *ipv6-address*} | **host** *host-name* | **host domain** *domain-name* | **user** *user-fqdn* | **user domain** *domain-name*}

**no match identity** {**group** *group-name* | **address** {*address* [*mask*] [*fvrf*] | **ipv6** *ipv6-address*} | **host** *host-name* | **host domain** *domain-name* | **user** *user-fqdn* | **user domain** *domain-name*}

## Syntax Description

| | |
|---|---|
| **group** *group-name* | A Unity group that matches identification (ID) type ID_KEY_ID. If Unity and main mode Rivest, Shamir, and Adelman (RSA) signatures are used, the *group-name* argument matches the Organizational Unit (OU) field of the Distinguished Name (DN). |
| **address** *address* [*mask*] [*fvrf*] | Identity that matches the identity of type ID_IPV4_ADDR.<br>• *mask*—Use to match the range of the address.<br>• *fvrf*—Use to match the address in the front door Virtual Route Forwarding (FVRF) Virtual Private Network (VPN) space. |
| **ipv6** *ipv6-address* | Identity that matches the identity of type ID_IPV6_ADDR. |
| **host** *host-name* | Identity that matches an identity of the type ID_FQDN. |
| **host domain** *domain-name* | Identity that matches an identity of the type ID_FQDN, whose fully qualified domain name (FQDN) ends with the domain name. |
| **user** *user-fqdn* | Identity that matches the FQDN. |
| **user domain** *domain-name* | Identity that matches the identities of the type ID_USER_FQDN. When the **user domain** keyword is present, all users having identities of the type ID_USER_FQDN and ending with "*domain-name*" will be matched. |

## Command Default

No default behavior or values

## Command Modes

ISAKMP profile configuration (conf-isa-prof)

## Command History

| Release | Modification |
|---|---|
| 12.2(15)T | This command was introduced. |
| 12.2(18)SXD | This command was integrated into Cisco IOS Release 12.2(18)SXD. |
| 12.4(4)T | The **ipv6** keyword and *ipv6-address a*rgument were added. |
| 12.2(33)SRA | This command was integrated into Cisco IOS release 12.(33)SRA. |
| Cisco IOS XE Release 2.1 | This command was introduced on Cisco ASR 1000 Series Routers. |

## Usage Guidelines

There must be at least one **match identity** command in an ISAKMP profile configuration. The peers are mapped to an ISAKMP profile when their identities are matched (as given in the ID payload of the Internet Key Exchange [IKE] exchange) against the identities that are defined in the ISAKMP profile. To uniquely map to an ISAKMP profile, no two ISAKMP profiles should match the same identity. If the peer identity is matched in two ISAKMP profiles, the configuration is invalid.

## Examples

The following example shows that the **match identity** command is configured:

```
crypto isakmp profile vpnprofile
 match identity group vpngroup
 match identity address 10.53.11.1
 match identity host domain example.com
 match identity host server.example.com
```

## Related Commands

| Command | Description |
|---|---|
| **crypto isakmp profile** | Defines an ISAKMP profile and audits IPSec user sessions. |

# match (IKEv2 policy)

To match a policy based on Front-door VPN Routing and Forwarding (FVRF) or local parameters, such as an IP address, use the **match** command in IKEv2 policy configuration mode. To delete a match, use the **no** form of this command.

**match address local** {*ipv4-address* | *ipv6-address* | **fvrf** *fvrf-name* | **any**}

**no match address local** {*ipv4-address* | *ipv6-address* | **fvrf** *fvrf-name* | **any**}

## Syntax Description

| | |
|---|---|
| **address local** | Matches a policy based on the local IPv4 or IPv6 address. |
| *ipv4-address* | IPv4 address. |
| *ipv6-address* | IPv6 address. |
| **fvrf** | Matches a policy based on the user-defined FVRF. |
| *frvf-name* | FVRF name |
| **any** | Matches a policy based on any FVRF. |

## Command Default

If no match address is specified, the policy matches all local addresses.

## Command Modes

IKEv2 policy configuration (crypto-ikev2-policy)

## Command History

| Release | Modification |
|---|---|
| 15.1(1)T | This command was introduced. |
| 15.1(4)M | This command was modified. Support was added for IPv6 addresses. |
| Cisco IOS XE Release 3.3S | This command was integrated into Cisco IOS XE Release 3.3S. |

## Usage Guidelines

Use this command to match a policy based on the FVRF or the local IP address (IPv4 or IPv6). The FVRF specifies the VRF in which the IKEv2 security association (SA) packets are negotiated. The default FVRF is the global FVRF. Use the **match fvrf any** command to match a policy based on any FVRF.

A policy with no match address local statement will match all local addresses. A policy with no match FVRF statement will match the global FVRF. If there are no match statements, an IKEv2 policy matches all local addresses in the global VRF.

## Examples

The following example shows how to match an IKEv2 policy based on the FVRF and the local IPv4 address:

```
Router(config)# crypto ikev2 policy policy1
Router(config-ikev2-policy)# proposal proposal1
Router(config-ikev2-policy)# match fvrf fvrf1
Router(config-ikev2-policy)# match address local 10.0.0.1
```

The following example shows how to match an IKEv2 policy based on the FVRF and the local IPv6 address:

```
Router(config)# crypto ikev2 policy policy1
Router(config-ikev2-policy)# proposal proposal1
Router(config-ikev2-policy)# match fvrf fvrf1
Router(config-ikev2-policy)# match address local 2001:DB8:0:ABCD::1
```

**Related Commands**

| Command | Description |
|---|---|
| **crypto ikev2 policy** | Defines an IKEv2 policy. |
| **proposal** | Specifies the proposals that must be used in the IKEv2 policy. |
| **show crypto ikev2 policy** | Displays the default or user-defined IKEv2 policy. |

# match (IKEv2 profile)

To match a profile on front-door VPN routing and forwarding (FVRF) or local parameters such as the IP address, the peer identity, or the peer certificate, use the **match** command in IKEv2 profile configuration mode. To delete a match, use the **no** form of this command.

**match** {**address local** {*ipv4-address* | *ipv6-address* | **interface** *name*} | **certificate** *certificate-map* | **fvrf** {*fvrf-name* | **any**} | **identity remote** {**address** {*ipv4-address* [*mask*] | *ipv6-address prefix*} | **email** [*domain*] *string* | **fqdn** [*domain*] *string* | **key-id** *opaque-string*}

**no match** {**address local** {*ipv4-address* | *ipv6-address* | **interface** *name*} | **certificate** *certificate-map*} | **fvrf** {*fvrf-name* | **any**} | **identity remote** {**address** {*ipv4-address* [*mask*] | *ipv6-address prefix*} | **email** [*domain*] *string* | **fqdn** [*domain*] *string* | **key-id** *opaque-string*}

## Syntax Description

| | |
|---|---|
| **address local** {*ipv4-address* \| *ipv6-address*} | Matches the profile based on the local IPv4 or IPv6 address. |
| **interface** *name* | Matches the profile based on the local interface. |
| **certificate** *certificate-map* | Matches the profile based on fields in the certificate received from the peer. |
| **fvrf** *fvrf-name* | Matches the profile based on the user-defined FVRF. The default FVRF is global. |
| **any** | Matches the profile based on any FVRF.<br>**Note** The **match vrf any** command must be explicitly configured to match all VRFs. |
| **identity remote** | Match a profile based on the remote IKEv2 identity field in the AUTH exchange. |
| **address** {*ipv4-address* [*mask*] \| *ipv6-address prefix*} | Matches a profile based on the identity of the type remote IPv4 address and its subnet mask or IPv6 address and its prefix length. |
| **key-id** *opaque-string* | Matches a profile based on the identity of the type remote key ID. |
| **email** | Matches a profile based on the identity of the type remote email ID. |
| **fqdn** *fqdn-name* | Matches a profile based on the identity of the type remote Fully Qualified Domain Name (FQDN). |
| **domain** *string* | Matches a profile based on the domain part of remote identities of the type FQDN or email. |

## Command Default

A match is not specified.

## Command Modes

IKEv2 profile configuration (crypto-ikev2-profile)

## Command History

| Release | Modification |
|---|---|
| 15.1(1)T | This command was introduced. |
| 15.1(4)M | This command was modified. Support was added for IPv6 addresses. |
| Cisco IOS XE Release 3.3S | This command was integrated into Cisco IOS XE Release 3.3S. |

## Usage Guidelines

In an IKEv2 profile, multiple match statements of the same type are logically ORed and match statements of different types are logically ANDed.

**Note** The **match identity remote** and **match certificate** statements are considered the same type of statements and are ORed.

The result of configuring multiple **match certificate** statements is the same as configuring one **match certificate** statement. Hence, using a single **match certificate** statement as a certificate map caters to multiple certificates and is independent of trustpoints.

**Note** There can only be one match FVRF statement.

For example, the following command translates to the subsequent “and”, “or” statement:

```
crypto ikev2 profile profile-1
 match vrf green
 match local address 10.0.0.1
 match local address 10.0.0.2
 match certificate remote CertMap
```

(vrf = green AND (local addr = 10.0.0.1 OR local addr = 10.0.0.1) AND remote certificate match CertMap).

There is no precedence between match statements of different types, and selection is based on the first match. Configuration of overlapping profiles is considered as a misconfiguration.

## Examples

The following examples show how an IKEv2 profile is matched on the remote identity. The following profile caters to peers that identify using **fqdn example.com** and authenticate with **rsa-signature** using **trustpoint-remote**. The local node authenticates with **pre-share** using **keyring-1**.

```
Router(config)# crypto ikev2 profile profile2
Router(config-ikev2-profile)# match identity remote fqdn example.com
Router(config-ikev2-profile)# identity local email router2@example.com
Router(config-ikev2-profile)# authentication local pre-share
Router(config-ikev2-profile)# authentication remote rsa-sig
Router(config-ikev2-profile)# keyring keyring-1
Router(config-ikev2-profile)# pki trustpoint trustpoint-remote verify
Router(config-ikev2-profile)# lifetime 300
Router(config-ikev2-profile)# dpd 5 10 on-demand
Router(config-ikev2-profile)# virtual-template 1
```

**Related Commands**

| Command | Description |
|---|---|
| **crypto ikev2 profile** | Defines an IKEv2 profile. |
| **identity (IKEv2 profile)** | Specifies how the local or remote router identifies itself to the peer and communicates with the peer in the RSA authentication exchange. |
| **authentication (IKEv2 profile)** | Specifies the local and remote authentication methods in an IKEv2 profile. |
| **keyring (IKEv2 profile)** | Specifies a locally defined or AAA-based keyring. |
| **pki trustpoint** | Specifies the router to use the PKI trustpoints in the RSA signature authentication. |

# match invalid-command

To locate invalid commands on a Post Office Protocol, Version 3 (POP 3) server or an Internet Message Access Protocol (IMAP) connection, use the **match invalid-command** in class-map configuration mode. To stop locating invalid commands, use the **no** form of this command.

**match invalid-command**

**no match invalid-command**

## Syntax Description

This command has no arguments or keywords.

## Command Default

It is not required that invalid commands be located.

## Command Modes

Class-map configuration

## Command History

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

## Usage Guidelines

You can use this command only after entering the **class-map type inspect imap** or **class-map type inspect pop3** command.

## Examples

The following example causes the Zone-Based Policy Firewall software to locate invalid commands on the POP3 server:

```
class-map type inspect pop3 pop3-class
 match invalid-command
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect imap** | Configures inspection parameters for IMAP. |
| **class-map type inspect pop3** | Configures inspection parameters for POP3. |

# match login clear-text

To find a nonsecure login when using an Internet Message Access Protocol (IMAP) or Post Office Protocol, Version 3 (POP3) server, use the **match login clear-text** command in class-map configuration mode. To disable this match criteria, use the **no** form of this command.

**match login clear-text**

**no match login clear-text**

## Syntax Description

This command has no arguments or keywords.

## Command Default

Finding non-secure logins is not required.

## Command Modes

Class-map configuration

## Command History

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

## Usage Guidelines

You can use this command either when you are configuring a POP3 firewall class map after you enter the **class-map type inspect pop3** command or when you are configuring an IMAP firewall class map after you enter the **class-map type inspect imap** command.

## Examples

The following example determines if the login process is happening in clear-text:

```
class-map type inspect pop3 pop3-class
 match login clear-text
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect imap** | Configures inspection parameters for IMAP. |
| **class-map type inspect pop3** | Configures inspection parameters for POP3. |
| **ip inspect name** | Defines a set of inspection rules. |

# match message

To configure the match criterion for a class map on the basis of H.323 protocol messages, use the **match message** command in class-map configuration mode. To remove the H.323-based match criterion from a class map, use the **no** form of this command.

**match message** *message-name*

**no match message** *message-name*

## Syntax Description

| | |
|---|---|
| *message-name* | Name of the message used as a message criterion. The supported message criteria are as follows:<br>• **alerting**—H.225 ALERTING message<br>• **call-proceeding**—H.225 CALL PROCEEDING message<br>• **connect**—H.225 CONNECT message<br>• **facility**—H.225 FACILITY message<br>• **release-complete**—H.225 RELEASE COMPLETE message<br>• **setup**—H.225 SETUP message<br>• **status**—H.225 STATUS message<br>• **status-enquiry**—H.225 STATUS ENQUIRY message |

## Command Default

None

## Command Modes

Class-map configuration (config-cmap)

## Command History

| Release | Modification |
|---|---|
| 12.4(20)T | This command was introduced. |

## Usage Guidelines

Use the **match message** command to inspect H.323 traffic based on the message criterion.

The **match message** command is available under the **class-map type inspect h323** command.

## Examples

The following example shows how to configure an H.323 specific class-map to match H.225 SETUP or H.225 RELEASE COMPLETE messages only.

```
class-map type inspect h323 match-any my_h323_rt_msgs
match message setup
match message release-complete
```

**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. |

# match mime content-type regex

To specify Multipurpose Internet Mail Extension (MIME) content file types, which are restricted in attachments in the body of the e-mail being sent over SMTP, use the **match mime content-type regex** command in class-map configuration mode. To disable this inspection parameter, use the **no** form of this command.

**match mime content-type regex** *content-type-regex*

**no match mime content-type regex** *content-type-regex*

Syntax Description

| | |
|---|---|
| *content-type-regex* | Specifies the type of content in the MIME header in regular expression form. |

**Command Default**

The content type regular expression is not defined.

**Command Modes**

Class-map configuration

**Command History**

| Release | Modification |
|---|---|
| 12.4(20)T | This command was introduced. |
| Cisco IOS XE Release 3.2S | This command was integrated into Cisco IOS XE Release 3.2S. |

**Usage Guidelines**

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

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

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

**Examples**

The following example shows how to configure an SMTP application firewall policy to specify that any form of JPEG image content be restricted in attachments in the body of the e-mail being sent over SMTP:

```
parameter-map type regex jpeg
 pattern "*image//*"

class-map type inspect smtp c1
 match mime content-type regex jpeg

policy-map type inspect smtp p1
 class type inspect smtp c1
 log
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect smtp** | Creates a class map for the SMTP protocol so that the match criteria is set to match criteria for this class map. |
| **class type inspect smtp** | Configures an SMTP class-map firewall for SMTP inspection parameters. |
| **parameter-map type regex** | Enters the parameter-map name of a specific traffic pattern. |
| **pattern** | Cisco IOS regular expression (regex) pattern that matches the traffic pattern for the e-mail sender or user accounts from suspected domains that are causing the spam e-mail. |
| **policy-map type inspect smtp** | Create a Layer 7 SMTP policy map. |

# match mime encoding

To restrict unknown Multipurpose Internet Mail Extension (MIME) content-encoding types or values from being transmitted over SMTP, use the **match mime encoding** command in class-map configuration mode. To disable this inspection parameter, use the **no** form of this command.

**match mime encoding** {**unknown** | *WORD* | *encoding-type*}

**no match mime encoding** {**unknown** | *WORD* | *encoding-type*}

## Syntax Description

| | |
|---|---|
| **unknown** | Specify this keyword if the content-transfer-encoding value in the e-mail does not match any of the ones in the list to restrict unknown and potentially dangerous encodings. |
| *WORD* | Specifies a user-defined content-transfer encoding type, which must begin with 'X' (example, "Xmyencodingscheme").<br>Non-alphanumeric characters, such as hyphens, are not supported. |
| *encoding-type* | Specifies one of the pre-configured content-transfer-encoding type:<br>– **7-bit**–ASCII characters<br>– **8-bit**–Facilitates the exchange of e-mail messages containing octets outside the 7-bit ASCII range.<br>– **base64**–Any similar encoding scheme that encodes binary data by treating it numerically and translating it into a base 64 representation.<br>– **quoted-printable**–Encoding using printable characters (i.e. alphanumeric and the equals sign "=") to transmit 8-bit data over a 7-bit data path. It is defined as a MIME content transfer encoding for use in Internet e-mail.<br>– **binary**–Representation for numbers using only two digits (usually, 0 and 1).<br>– **x-uuencode**–Nonstandard encoding.<br>• The **quoted-printable** and **base64** encoding types tell the email client that a binary-to-text encoding scheme was used and that appropriate initial decoding is necessary before the message can be read with its original encoding. |

## Command Default

The MIME encoding type or value is not defined.

## Command Modes

Class-map configuration

## Command History

| Release | Modification |
|---|---|
| 12.4(20)T | This command was introduced. |
| Cisco IOS XE Release 3.2S | This command was integrated into Cisco IOS XE Release 3.2S. |

## Usage Guidelines

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

## Examples

The following example shows how to configure an SMTP application firewall policy to specify that any quoted-printable encoding field in the MIME header within the SMTP body be restricted in e-mail being sent over SMTP:

```
class-map type inspect smtp c1
 match mime encoding quoted-printable

policy-map type inspect smtp p1
 class type inspect smtp c1
 log
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect smtp** | Creates a class map for the SMTP protocol so that the match criteria is set to match criteria for this class map. |
| **class type inspect smtp** | Configures an SMTP class-map firewall for SMTP inspection parameters. |
| **log** | Generates a log of messages. |
| **policy-map type inspect smtp** | Create a Layer 7 SMTP policy map. |

# match program-number

To specify the allowed Remote Procedure Call (RPC) protocol program number as a match criterion, use the **match program-number** command in class-map configuration mode. To disable this match criterion, use the **no** form of this command.

**match program-number** *program-number*

**no match program-number** *program-number*

## Syntax Description

| | |
|---|---|
| *program-number* | Allowed program number. |

## Command Default

Disabled

## Command Modes

Class-map configuration

## Command History

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

## Usage Guidelines

This match criterion is allowed only for SUN Remote Procedure Call (SUNRPC) class maps. You can use the **match program-number** command only after specifying the **class-map type inspect sunrpc** command.

## Examples

The following example configures the program number 2345 as a match criterion in the class map rpc-prog-nums:

```
class-map type inspect sunrpc rpc-prog-nums
 match program-number 2345
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect sunrpc** | Configures inspection parameters for SUNRPC. |
| **ip inspect name** | Defines a set of inspection rules. |

# match protocol (zone)

To configure the match criterion for a class map on the basis of the specified protocol, use the **match protocol** command in class-map configuration mode. To remove the protocol-based match criterion from a class map, use the **no** form of this command.

**match protocol** *protocol-name* [*parameter-map*] [**signature**]

**no match protocol** *protocol-name* [*parameter-map*] [**signature**]

## Syntax Description

| | |
|---|---|
| *protocol-name* | Name of the protocol used as a matching criterion.<br>For a list of supported protocols, use the CLI help option (**?**) on your platform. |
| *parameter-map* | (Optional) Protocol-specific parameter map. |
| **signature** | (Optional) Enables signature-based classification for peer-to-peer (P2P) packets.<br>**Note** This option is available only for P2P traffic. |

## Command Default

No protocol-based match criterion for a class map is configured.

## Command Modes

class-map configuration (config-cmap)

## Command History

| Release | Modification |
|---|---|
| 12.4(6)T | This command was introduced for the zone-based policy firewall. |
| 12.4(9)T | This command was modified. Support for the following protocols was added:<br>• P2P protocols: **bittorrent**, **directconnect**, **edonkey**, **fasttrack**, **gnutella**, **kazaa2**, and **winmx**<br>• Instant Messenger (IM) protocols: **aol**, **msnmsgr**, and **ymsgr**<br>Also, the **signature** keyword was added to be used only with P2P protocols. |
| 12.4(11)T | This command was modified. Support for the H.225 Remote Access Services (RAS) protocol and the **h225ras** keyword was added. |
| 12.4(20)T | This command was modified. Support for the I Seek You (ICQ) and Windows Messenger IM protocols and the following keywords was added: **icq**, **winmsgr**<br>Support for the H.323 protocol and the **h323** keyword was added.<br>Support for the Session Initiation Protocol (SIP) protocol and the **sip** keyword was added. |
| Cisco IOS XE Release 2.4 | This command was integrated into Cisco IOS XE Release 2.4. |

| Release | Modification |
|---|---|
| 15.0(1)M | This command was modified. The **extended** keyword was removed from the protocol name. |
| 15.1(1)T | This command was modified. Support for the CU-SeeMe protocol and **cuseeme** keyword was removed. |
| 15.0(1)S | This command was integrated into Cisco IOS Release 15.0(1)S. The following keywords were added: **netbios-dgm**, **netbios-ns**, and **netbios-ssn**. |

## Usage Guidelines

Use the **match protocol** command to specify the traffic based on a particular protocol. You can use this command in conjunction with the **match access-group** and **match class-map** commands to build sophisticated traffic classes.

The **match protocol** command is available under the **class-map type inspect** command.

If you enter the **match protocol** command under the **class-map type inspect** command, the Port to Application Mappings (PAM) are honored when the protocol field in the packet is matched against this command. All the port mappings configured in the PAM table appear under the class map.

When packets are matched to a protocol, a traffic rate is generated for these packets. In a zone-based firewall policy, only the first packet that creates a session matches the policy. Subsequent packets in this flow do not match the filters in the configured policy, but instead match the session directly. The statistics related to subsequent packets are shown as part of the inspect action.

In Cisco IOS Release 12.4(15)T only, if Simple Mail Transfer Protocol (SMTP) is currently configured for inspection in a class map and the inspection of Extended SMTP (ESMTP) needs to be configured, then the **no match protocol smtp** command must be entered before adding the **match protocol smtp extended** command. To revert to regular SMTP inspection, use the **no match protocol smtp extended** command and then enter the **match protocol smtp** command.

In Cisco IOS Release 12.4(15)T, if these commands are not configured in the proper order, then the following error displays:

```
%Cannot add this filter.Remove match protocol smtp filter and then add this filter
```

In Cisco IOS Release 15.0(1)M and later releases, the **extended** keyword was removed from the **match protocol smtp** command.

## Examples

The following example shows how to specify a class map called c1 and configure the HTTP protocol as a match criterion:

```
class-map type inspect c1
 match protocol http
```

The following example shows how to specify different class maps for ICQ and Windows Messenger IM applications:

```
! Define the servers for ICQ.
parameter-map type protocol-info icq-servers
 server name *.icq.com snoop
 server name oam-d09a.blue.aol.com

! Define the servers for Windows Messenger.
parameter-map type protocol-info winmsgr-servers
 server name messenger.msn.com snoop
```

```
! Define servers for yahoo.
parameter-map type protocol-info yahoo-servers
 server name scs*.msg.yahoo.com snoop
 server name c*.msg.yahoo.com snoop

! Define class-map to match ICQ traffic.
class-map type inspect icq-traffic
 match protocol icq icq-servers

! Define class-map to match windows Messenger traffic.
class-map type inspect winmsgr-traffic
 match protocol winmsgr winmsgr-servers
!

! Define class-map to match text-chat for windows messenger.
class-map type inspect winmsgr winmsgr-textchat
 match service text-chat
!

Define class-map to match default service
class-map type inspect winmsgr winmsgr-defaultservice
 match service any
!
```

The following example shows how to specify a class map called c1 and configure the netbios-dgm protocol as a match criterion:

```
class-map type inspect c1
 match protocol netbios-dgm
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect** | Creates a Layer 3 or Layer 4 inspect type class map. |
| **match access-group** | Configures the match criteria for a class map based on the specified ACL. |

# match protocol h323-annexe

To enable the inspection of H.323 protocol Annex E traffic which works on the User Datagram Protocol (UDP) diagnostic port or TCP port 2517, use the **match protocol h323-annexe** command in class-map configuration mode. To disable the inspection, use the **no** form of this command.

**match protocol h323-annexe**

**no match protocol h323-annexe**

## Syntax Description

This command has no arguments or keywords.

## Command Default

None

## Command Modes

Class-map configuration (config-cmap)

## Command History

| Release | Modification |
|---|---|
| 12.4(20)T | This command was introduced. |

## Usage Guidelines

Use the **match protocol h323-annexe** command to inspect traffic based on Annex E of the H.323 protocol that uses the UDP diagnostic port or TCP port 2517. You can use this command in conjunction with the **match access-group** command to build sophisticated traffic classes.

The **match protocol h323-annexe** command is available under the **class-map type inspect** command.

## Examples

The following example shows how to configure a voice policy to inspect the H.323 protocol Annex E packets for the “my-voice-class” class map.

```
class-map type inspect match-all my-voice-class
 match protocol h323-annexe
```

## 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. |
| **match access-group** | Configures the match criteria for a class map based on the specified ACL. |
| **match protocol h323-nxg** | Enables the inspection of H.323 protocol Annex G traffic exchanged between border elements (BE) using the User Datagram Protocol (UDP) diagnostic port or TCP port 2099. |

# match protocol h323-nxg

To enable the inspection of H.323 protocol Annex G traffic exchanged between border elements (BE) using User Datagram Protocol (UDP) diagnostic port or TCP port 2099, use the **match protocol h323-nxg** command in class-map configuration mode. To disable the inspection, use the **no** form of this command.

**match protocol h323-nxg**

**no match protocol h323-nxg**

## Syntax Description

This command has no arguments or keywords.

## Command Default

None

## Command Modes

Class-map configuration (config-cmap)

## Command History

| Release | Modification |
|---|---|
| 12.4(20)T | This command was introduced. |

## Usage Guidelines

Use the **match protocol h323-nxg** command to inspect traffic based on Annex G of the H.323 protocol that uses the UDP diagnostic port or TCP port 2099 to exchange traffic between border elements. You can use this command in conjunction with the **match access-group** command to build sophisticated traffic classes.

The **match protocol h323-nxg** command is available under the **class-map type inspect** command.

## Examples

The following example shows how to configure a voice policy to inspect the H.323 protocol Annex G packets for the “my-voice-class” class map.

```
class-map type inspect match-all my-voice-class
 match protocol h323-nxg
```

## 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. |
| **match access-group** | Configures the match criteria for a class map based on the specified ACL. |
| **match protocol h323-annexe** | Enables the inspection of H.323 protocol Annex E traffic which works on the UDP diagnostic port or TCP Port 2517. |

# match protocol-violation

To configure a Session Initiation Protocol (SIP) class map to use the protocol-violation method as a match criterion for permitting or denying SIP traffic, use the **match protocol-violation** command in class-map configuration mode. To remove the protocol-violation based match criterion from a class map, use the **no** form of this command.

**match protocol-violation**

**no match protocol-violation**

## Syntax Description

This command has no arguments or keywords.

## Command Default

No match criterion is configured.

## Command Modes

Class-map configuration (config-cmap)

## 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 this command when configuring an SIP firewall class map, after entering the **class-map type inspect** command.

## Examples

The following example shows how to specify the protocol-violation method as a match criterion.

```
Router(config)# class-map type inspect sip sip-class
Router(config-cmap)# match protocol-violation
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect sip** | Creates a class map for SIP. |

# match recipient address regex

To specify a non-existent e-mail recipient pattern in order to learn a spam sender and their domain information by luring them to use this contrived e-mail recipient, use the **match recipient address regex** command in class-map configuration mode. To disable this inspection parameter, use the **no** form of this command.

**match recipient address regex** *parameter-map-name*

**no match recipient address regex** *parameter-map-name*

## Syntax Description

| | |
|---|---|
| *parameter-map-name* | Specifies the name of the non-existent e-mail recipient pattern. |

## Command Default

The fictitious names of e-mail recipients are not defined.

## Command Modes

Class-map configuration

## Command History

| Release | Modification |
|---|---|
| 12.4(20)T | This command was introduced. |
| Cisco IOS XE Release 3.2S | This command was integrated into Cisco IOS XE Release 3.2S. |

## Usage Guidelines

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

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

## Examples

The following example shows how to configure a regular expression non-existent e-mail recipient pattern:

```
parameter-map type regex known-unknown-users
 pattern "john@mydomain.com"

class-map type inspect smtp c1
 match recipient address regex known-unknown-users

policy-map type inspect smtp p1
 class type inspect smtp c1
 reset
```

**Related Commands**

| Command | Description |
|---|---|
| **class-map type inspect smtp** | Creates a class map for the SMTP protocol so that the match criteria is set to match criteria for this class map. |
| **class type inspect smtp** | Configures an SMTP class-map firewall for SMTP inspection parameters. |
| **parameter-map type regex** | Enters the parameter-map name of a specific traffic pattern. |
| **pattern** | Cisco IOS regular expression (regex) pattern that matches the traffic pattern for the e-mail sender or user accounts from suspected domains that are causing the spam e-mail. |
| **policy-map type inspect smtp** | Create a Layer 7 SMTP policy map. |
| **reset** | (Optional) Drops an SMTP connection with an SMTP sender (client) if it violates the specified policy. This action sends an error code to the sender and closes the connection gracefully. |

# match recipient count gt

To specify an action that occurs when a number of invalid recipients appear on an SMTP connection, use the **match recipient count gt** command in class-map configuration mode. To disable this inspection parameter, use the **no** form of this command.

**match recipient count gt** *value*

**no match recipient count gt** *value*

## Syntax Description

| | |
|---|---|
| *value* | Specifies the number of RCPT SMTP commands sent by the sender (client) to recipients who are specified in a single SMTP transaction to limit these commands. |

## Command Default

The number of RCPT SMTP commands sent by a sender to recipients is not defined.

## Command Modes

Class-map configuration

## Command History

| Release | Modification |
|---|---|
| 12.4(20)T | This command was introduced. |
| Cisco IOS XE Release 3.2S | This command was integrated into Cisco IOS XE Release 3.2S. |

## Usage Guidelines

Spammers who search for a large number of user accounts in a domain typically send the same e-mail to all the user accounts they find in this domain. Spammers can be identified and restricted from searching for user accounts in a domain by using the **match recipient count gt** command.

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

## Examples

The following example shows how to configure an SMTP application firewall policy to determine the number of **RCPT** lines and invalid recipients, for which the server has replied "500 No such address," in the SMTP transaction:

```
class-map type inspect smtp c1
 match recipient count gt 25

policy-map type inspect smtp p1
 class type inspect smtp c1
 reset
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect smtp** | Creates a class map for the SMTP protocol so that the match criteria is set to match criteria for this class map. |
| **class type inspect smtp** | Configures an SMTP class-map firewall for SMTP inspection parameters. |
| **policy-map type inspect smtp** | Create a Layer 7 SMTP policy map. |
| **reset** | (Optional) Drops an SMTP connection with an SMTP sender (client) if it violates the specified policy. This action sends an error code to the sender and closes the connection gracefully. |

# match recipient invalid count gt

To identify and restrict the number of invalid SMTP recipients that can appear in an e-mail from senders who try common names on a domain in the hope that they discover a valid user name to whom they can send spam, use the **match recipient invalid count gt** command in class-map configuration mode. To disable this inspection parameter, use the **no** form of this command.

**match recipient invalid count gt** *value*

**no match sender address regex** *value*

## Syntax Description

| | |
|---|---|
| *value* | Specifies a maximum number of invalid e-mail recipients on this SMTP connection. |

## Command Default

The a number of invalid e-mail recipients is not defined.

## Command Modes

Class-map configuration

## Command History

| Release | Modification |
|---|---|
| 12.4(20)T | This command was introduced. |
| Cisco IOS XE Release 3.2S | This command was integrated into Cisco IOS XE Release 3.2S. |

## Usage Guidelines

If a sender specifies in an invalid e-mail recipient and SMTP encounters this invalid recipient on the SMTP connection, then SMTP sends an error code reply to the e-mail sender (client) to specify another recipient. In this case, the event did not violate the SMTP protocol or indicate that this particular SMTP connection is bad. However, if a pattern of invalid recipients appears, then a reasonable threshold can be set to restrict these nuisance SMTP connections.

## Examples

The following example shows how to configure an SMTP application firewall policy that restricts the number of invalid e-mail recipients on this SMTP connection to 5:

```
class-map type inspect smtp c1
 match recipient invalid count gt 5

policy-map type inspect smtp p1
 class type inspect smtp c1
 reset
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect smtp** | Creates a class map for the SMTP protocol so that the match criteria is set to match criteria for this class map. |
| **class type inspect smtp** | Configures an SMTP class-map firewall for SMTP inspection parameters. |
| **policy-map type inspect smtp** | Create a Layer 7 SMTP policy map. |
| **reset** | (Optional) Drops an SMTP connection with an SMTP sender (client) if it violates the specified policy. This action sends an error code to the sender and closes the connection gracefully. |

# match reply ehlo

To identify and mask a service extension parameter in the EHLO server reply (e.g. 8BITMIME, ETRN) to prevent a sender (client) from using that particular service extension, use the **match reply ehlo** command in class-map configuration mode. To disable this inspection parameter, use the **no** form of this command.

**match reply ehlo** {*parameter* | *WORD*}

**no match reply ehlo** {*parameter* | *WORD*}

## Syntax Description

| | |
|---|---|
| *parameter* | Specify a parameter from the well-known EHLO keywords. |
| *WORD* | Specify an extension which is not on the EHLO list (e.g. private extension XFOOBAR).<br>Non-alphanumeric characters, such as hyphens, are not supported. |

## Command Default

The service extension parameter in the EHLO server reply is not defined or masked.

## Command Modes

Class-map configuration

## Command History

| Release | Modification |
|---|---|
| 12.4(20)T | This command was introduced. |
| Cisco IOS XE Release 3.2S | This command was integrated into Cisco IOS XE Release 3.2S. |

## Examples

The following example shows how to configure an SMTP application firewall policy that identifies and masks a well-known service extension parameter in the EHLO server reply:

```
class-map type inspect smtp c1
 match reply ehlo ETRN

policy-map type inspect smtp p1
 class type inspect smtp c1
 log
 mask
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect smtp** | Creates a class map for the SMTP protocol so that the match criteria is set to match criteria for this class map. |
| **class type inspect smtp** | Configures an SMTP class-map firewall for SMTP inspection parameters. |
| **log** | Logs an action related to this class-type in the SMTP policy map. |

| Command | Description |
|---|---|
| **mask (policy-map)** | Explicitly masks specified SMTP commands or the parameters returned by the server in response to an EHLO command. |
| **policy-map type inspect smtp** | Create a Layer 7 SMTP policy map. |

# match req-resp

To configure a Session Initiation Protocol (SIP) class map to use the req-resp methods as a match criterion for permitting or denying SIP traffic, use the **match req-resp** command in class-map configuration mode. To remove the req-resp based match criterion from a class map, use the **no** form of this command.

**match req-resp header** *field* **regex** *regex-parameter-map*

**no match req-resp header** *field* **regex** *regex-parameter-map*

## Syntax Description

| | |
|---|---|
| **header** | Identifies the SIP header field. |
| *field* | Name of the request header field. The following are valid request header fields: **accept**, **accept-encoding**, **accept-language**, **alert-info**, **allow**, **contact**, **content-disposition**, **content-encoding**, **content-language**, **content-length**, **content-type**, **from**, **record-route**, **supported**, **to**, **user-agent**, **via**. |
| **regex** | Indicates that a regular expression will follow. |
| *regex-parameter-map* | Configures a parameter map of type **regex**. |

## Command Default

No match criterion is configured.

## Command Modes

Class-map configuration (config-cmap)

## 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 this command when configuring an SIP firewall class map, after entering the **class-map type inspect** command.

## Examples

The following example shows how to specify the req-resp method as a match criterion.

```
Router(config)# class-map type inspect sip sip-class
Router(config-cmap)# match req-resp header via regex unsecure_proxy
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect sip** | Creates a class map for SIP. |

# match req-resp body length

To configure an HTTP class map to use the minimum or maximum message size, in bytes, as a match criterion for permitting or denying HTTP traffic through the firewall, use the **match req-resp body length** command in class-map configuration mode. To remove message-size limitations from your configuration, use the **no** form of this command.

**match req-resp body length** {**lt** *bytes* | **gt** *bytes*}

**no match req-resp body length** {**lt** *bytes* | **gt** *bytes*}

## Syntax Description

| | |
|---|---|
| **lt** *bytes* | Minimum number of bytes in each message. The range is from 0 to 65535. |
| **gt** *bytes* | Message cannot be greater than the specified number of bytes. |

## Command Default

Message size is not considered when permitting or denying HTTP messages.

## Command Modes

Class-map configuration

## Command History

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

## Usage Guidelines

You can use this command when you are configuring an HTTP firewall policy map,only after entering the **class-map type inspect http** command.

If the message body length is less than or greater than the specified values, a match occurs.

## Examples

The following example, which shows how to define the HTTP application firewall policy http-class, will not permit HTTP messages longer than 1 byte:

```
class-map type inspect http http-class
 match req-resp body length 1
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect http** | Creates a class map for HTTP. |

# match req-resp header content-type

To match traffic based on the content type of the HTTP body, use the **match req-resp header content-type** command in class-map configuration mode. To disable this inspection parameter, use the **no** form of this command.

**match req-resp header content-type** {**violation** | **mismatch** | **unknown**}

**no match req-resp header content-type** {**violation** | **mismatch** | **unknown**}

## Syntax Description

| | |
|---|---|
| **violation** | Flags a match if the content-type definition and the content type of the actual body do not match. |
| **mismatch** | Verifies the content-type of the response message against the accept field value of the request message. |
| **unknown** | Flags a match when an unknown content-type is found. |

## Command Default

No content-type checking is performed.

## Command Modes

Class-map configuration

## Command History

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

## Usage Guidelines

You can use the **match req-resp header content-type** command when you are configuring an HTTP firewall policy map, only after entering the **class-map type inspect http** command.

The **match req-resp header content-type** command configures a policy based on the content type of HTTP traffic. The command verifies that the header is one of the following supported content types:

- audio/*
- audio/basic
- audio/midi
- audio/mpeg
- audio/x-adpcm
- audio/x-aiff
- audio/x-ogg
- audio/x-wav
- application/msword
- application/octet-stream
- application/pdf
- application/postscript

- application/vnd.ms-excel
- application/vnd.ms-powerpoint
- application/x-gzip
- application/x-java-arching
- application/x-java-xm
- application/zip
- image/*
- image/cgf
- image/gif
- image/jpeg
- image/png
- image/tiff
- image/x-3ds
- image/x-bitmap
- image/x-niff
- image/x-portable-bitmap
- image/x-portable-greymap
- image/x-xpm
- text/*
- text/css
- text/html
- text/plain
- text/richtext
- text/sgml
- text/xmcd
- text/xml
- video/*
- video/-flc
- video/mpeg
- video/quicktime
- video/sgi
- video/x-avi
- video/x-fli
- video/x-mng
- video/x-msvideo

## Examples

The following example configures an HTTP class map based on the content type of HTTP traffic:

```
class-map type inspect http http-class
 match req-resp header content-type unknown
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect http** | Creates a class map for HTTP. |
| **content-type-verification** | Permits or denies HTTP traffic through the firewall on the basis of content message type. |
| **content-type-verification-match-req-rsp** | Verifies the content type of the HTTP response against the accept field of the HTTP request. |

# match req-resp header transfer-encoding

To permit or deny HTTP traffic according to the specified transfer encoding of the message, use the **match req-resp header transfer-encoding** command in class-map configuration mode. To remove this match criterion, use the **no** form of this command.

**match req-resp header transfer-encoding** {**chunked** | **compress** | **deflate** | **gzip** | **identity** | **all**}

**no match req-resp header transfer-encoding** {**chunked** | **compress** | **deflate** | **gzip** | **identity** | **all**}

## Syntax Description

| | |
|---|---|
| **chunked** | Encoding format (specified in RFC 2616, Hypertext Transfer Protocol—HTTP/1) in which the body of the message is transferred in a series of chunks; each chunk contains its own size indicator. |
| **compress** | Encoding format produced by the UNIX compress utility. |
| **deflate** | ZLIB format defined in RFC 1950, ZLIB Compressed Data Format Specification Version 3.3, combined with the deflate compression mechanism described in RFC 1951, DEFLATE Compressed Data Format Specification Version 1.3. |
| **gzip** | Encoding format produced by the gzip (GNU zip) program. |
| **identity** | Default encoding, which indicates that no encoding has been performed. |
| **all** | All of the transfer encoding types. |

## Command Default

None

## Command Modes

Class-map configuration

## Command History

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

## Usage Guidelines

You can use this command when you are configuring an HTTP firewall policy map, after entering the **class-map type inspect http** command.

## Examples

The following example permits or denies HTTP traffic according to the encoding format produced by the UNIX compress utility:

```
class-map type inspect http http-class
 match req-resp header transfer-encoding compress
```

**Related Commands**

| Command | Description |
|---|---|
| **class-map type inspect http** | Creates a class map for HTTP. |
| **transfer-encoding type** | Permits or denies HTTP traffic according to the specified transfer-encoding of the message. |

# match req-resp protocol-violation

To allow HTTP messages to pass through the firewall or to reset the TCP connection when HTTP noncompliant traffic is detected, use the **match req-resp protocol-violation** command in class-map configuration mode. To disable configured settings, use the **no** form of this command.

**match req-resp protocol-violation**

**no match req-resp protocol-violation**

## Syntax Description

This command has no arguments or keywords.

## Command Default

All traffic is allowed through the firewall.

## Command Modes

Class-map configuration

## Command History

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

## Usage Guidelines

You can use this command when you are configuring an HTTP firewall policy map, after entering the **class-map type inspect http** command.

The **match req-resp protocol-violation** command allows HTTP messages to pass through the firewall, If desired, in the policy map you can reset the TCP connection when HTTP noncompliant traffic is detected.

## Examples

The following example allows HTTP messages to pass through the firewall:

```
class-map type inspect http http-class
 match req-resp protocol-violation
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect http** | Creates a class map for HTTP. |

# match request

To configure a Session Initiation Protocol (SIP) class map to use the request methods as a match criterion for permitting or denying SIP traffic, use the **match request** command in class-map configuration mode. To remove request based match criterion from a class map, use the **no** form of this command.

**match request** {**method** *method-name* | **header** *field* **regex** *regex-parameter-map*}

**no match request** {**method** *method-name* | **header** *field* **regex** *regex-parameter-map*}

## Syntax Description

| | |
|---|---|
| **method** | Identifies the SIP request method. |
| *method-name* | Name of the method (for example, **ack**) used as a matching criterion. See the "Usage Guidelines" for a list of methods supported by most routers. |
| **header** | Identifies the SIP header field. |
| *field* | Name of the request header field. The following are valid request header fields: **accept**, **accept-encoding**, **accept-language**, **alert-info**, **allow**, **authorization**, **contact**, **content-disposition**, **content-encoding**, **content-language**, **content-length**, **content-type**, **from**, **in-reply-to**, **max-forwards**, **priority**, **proxy-authorization**, **proxy-require**, **record-route**, **route**, **subject**, **supported**, **to**, **user-agent**, **via**, **warning**. |
| **regex** | Indicates that a regular expression will follow. |
| *regex-parameter-map* | Configures a parameter map of type **regex**. |

## Command Default

No match criterion is configured.

## Command Modes

Class-map configuration (config-cmap)

## 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 this command when configuring an SIP firewall class map, after entering the **class-map type inspect** command.

### Supported Methods

Table 42 lists the request methods supported by most routers. For a complete list of supported methods, see the online help for the **match request** command on the router that you are using.

*Table 42 Supported Methods*

| Method Name | Description |
|---|---|
| **ack** | Acknowledges that the previous message is valid and accepted. |
| **bye** | Signifies intent to terminate a call. |
| **cancel** | Terminates any pending request. |
| **info** | Communicates midsession signaling information along the signaling path for a call. |
| **invite** | Sets up a call. |
| **message** | Sends an instant message. |
| **notify** | Informs subscribers of state changes. |
| **options** | Allows a user-agent (UA) to query another UA or a proxy server about its capabilities. |
| **prack** | Provides reliable transfer of provisional response messages. |
| **refer** | Indicates that the recipient should contact a third party using the contact information provided in the request. |
| **register** | Includes a contact address to which SIP requests for the address-of-record should be forwarded. |
| **subscribe** | Requests state subscription. It is a dialog creating method. |
| **update** | Allows a client to update the parameters of a session (for example, the set of media streams and their codecs), but has no impact on the state of a dialog. |

## Examples

The following example shows how to specify the request method **subscribe** as a match criterion.

```
Router(config)# class-map type inspect sip sip-class
Router(config-cmap)# match request method subscribe
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect sip** | Creates a class map for SIP. |

# match request length

To configure an HTTP firewall policy to use the uniform resource identifier (URI) or argument length in the request message as a match criterion for permitting or denying HTTP traffic, use the **match request length** command in class-map configuration mode. To remove this match criterion, use the **no** form of this command.

**match request** {**uri** | **arg**} **length gt** *bytes*

**no match request** {**uri** | **arg**} **length gt** *bytes*

## Syntax Description

| | |
|---|---|
| **uri** \| **arg** | Firewall will search the URI or argument length of the request message as the match criterion. |
| **gt** *bytes* | Permits HTTP traffic if the URL in the request message contains more than the specified number of bytes. |

## Command Default

URI or argument lengths are not considered when permitting or denying HTTP traffic.

## Command Modes

Class-map configuration

## Command History

| Release | Modification |
|---|---|
| 12.4(6)T | This command was introduced. |
| 12.4(9)T | The **arg** keyword was added. |

## Usage Guidelines

Use the **match request length** command to verify the length of the URI or argument that is being sent in a request message and apply the configured action when the length exceeds the configured threshold.

If a match is found, possible actions that can be specified within the policy are as follows: allow, reset, or log. (The log action triggers a syslog message when a match is found.)

## Examples

The following example shows how to configure an HTTP application firewall policy to raise an alarm whenever the URI length of a request message exceeds 3076 bytes:

```
class-map type inspect http uri_len_cm
 match request uri length gt 3076

policy-map type inspect http uri_len_pm
 class type inspect http uri_len_cm
 log
```

The following example shows how to configure an HTTP application firewall policy to raise an alarm whenever the argument length of a request message exceeds 512 bytes.

```
class-map type inspect http arg_len_cm
 match request arg length gt 512

policy-map type inspect http arg_len_pm
 class type inspect http arg_len_cm
 log
```

# match request method

To configure an HTTP class map to use the request methods or the extension methods as a match criterion for permitting or denying HTTP traffic, use the **match request method** command in class-map configuration mode. To remove this match criterion, use the **no** form of this command.

**match request method** {**connect** | **copy** | **delete** | **edit** | **get** | **getattribute** | **getattributenames** | **getproperties** | **head** | **index** | **lock** | **mkdir** | **move** | **options** | **post** | **put** | **revadd** | **revlabel** | **revlog** | **revnum** | **save** | **setattribute** | **startrev** | **stoprev** | **trace** | **unedit** | **unlock**}

**no match request method** {**connect** | **copy** | **delete** | **edit** | **get** | **getattribute** | **getattributenames** | **getproperties** | **head** | **index** | **lock** | **mkdir** | **move** | **options** | **post** | **put** | **revadd** | **revlabel** | **revlog** | **revnum** | **save** | **setattribute** | **startrev** | **stoprev** | **trace** | **unedit** | **unlock**}

## Syntax Description

| | |
|---|---|
| **connect** | Connect method. |
| **copy** | Copy extension method. |
| **delete** | Delete method. |
| **edit** | Edit extension method. |
| **get** | Get method. |
| **getattribute** | Getattribute extension method. |
| **getattributenames** | Getattributenames extension method. |
| **getproperties** | Getproperties method. |
| **head** | Head method. |
| **index** | Index extension method. |
| **lock** | Lock extension method. |
| **mkdir** | Mkdir extension method. |
| **move** | Move extension method. |
| **options** | Options method. |
| **post** | Post method. |
| **put** | Put method. |
| **revadd** | Revadd extension method. |
| **revlabel** | Revlabel extension method. |
| **revlog** | Revlog extension method. |
| **revnum** | Revnum extension method. |
| **save** | Save extension method. |
| **setattribute** | Setattribute extension method. |
| **startrev** | Startrev extension method. |
| **stoprev** | Stoprev extension method. |
| **trace** | Trace method. |
| **unedit** | Unedit extension method. |
| **unlock** | Unlock extension method. |

**Command Default** None

**Command Modes** Class-map configuration

**Command History**

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

**Usage Guidelines** You can use this command when you are configuring an HTTP firewall class map, after entering the **class-map type inspect http** command.

**Examples** The following example specifies that the match criteria is connect:

```
class-map type inspect http http-class
 match request method connect
```

**Related Commands**

| Command | Description |
|---|---|
| **class-map type inspect http** | Creates a class map for HTTP. |

# match request not regex

To negate a match result in a HTTP firewall policy, use the **match request not regex** command in class-map configuration mode. To reset the match criterion, use the **no** form of this command.

**match request not uri regex** *parameter-map-name*

**no match request not uri regex** *parameter-map-name*

## Syntax Description

| | |
|---|---|
| **uri** | Firewall policy will search the URI or argument as the match criterion. |
| *parameter-map-name* | HTTP-based parameter map as specified via the **parameter-map type** command. |

## Command Default

Match negation is not enabled.

## Command Modes

Class-map configuration (config-cmap)#

## Command History

| Release | Modification |
|---|---|
| 15.1(1)T | This command was introduced. |

## Usage Guidelines

Use the **match request not uri regex** command to negate a match result.

## Examples

The following example shows how to negate a match result and the output of the configuration in the running configuration.

```
Router(config)# policy-map type inspect http httppmap
Router(config-cmap)# match not request uri regex pmap
Router(config-cmap)# match request method post
Route(config-pmap)# class type inspect http cmap
Router(config-pmap-c)# reset
Router(config-pmap-c)# log
```

In the following configuration, if the HTTP POST request does not match the URL regular expression, It will be classified under class “httpcmap” and firewall will RESET the connection as it has RESET configured for this class.

```
parameter-map type regex pmap
 pattern .*Publications/OrderHardcopies/tabid/123/Default.aspx

class-map type inspect http match-all httpcmap
 match not request uri regex pmap
 match request method post

policy-map type inspect http pmap
 class type inspect http httpcmap
   reset
   log
```

```
class class-default
```

**Related Commands**

| Command | Description |
|---|---|
| **parameter-map type** | Defines a parameter map. |
| **class-map type inspect** | Defines an inspect type class map. |
| **match request regex** | Defines a HTTP firewall policy to permit or deny HTTP traffic. |
| **policy-map type inspect** | Defines an inspect type policy map. |

# match request port-misuse

To identify applications misusing HTTP port, use the **match request port-misuse** command in class-map configuration mode. To remove this inspection parameter, use the **no** form of this command.

**match request port-misuse** {**im** | **p2p** | **tunneling** | **any**}

**no match request port-misuse** {**im** | **p2p** | **tunneling** | **any**}

## Syntax Description

| | |
|---|---|
| **im** | Instant messaging protocol applications subject to inspection. |
| **p2p** | Peer-to-peer protocol applications subject to inspection. |
| **tunneling** | Tunneling applications subject to inspection: HTTPPort/HTTPHost. |
| **any** | Any type of misuse (**im**, **p2p**, and **tunneling**). |

## Command Default

Applications that are misusing the HTTP port cannot be identified.

## Command Modes

Class-map configuration

## Command History

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

## Usage Guidelines

You can use this command only after entering the **class-map type inspect http** command.

## Examples

The following example identifies all types of misuse of the HTTP port:

```
class-map type inspect http http-class
 match request port-misuse any
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect http** | Creates a class map for HTTP. |
| **port-misuse** | Permits or denies HTTP traffic through the firewall on the basis of specified applications in the HTTP message. |

# match request regex

To configure an HTTP firewall policy to permit or deny HTTP traffic on the basis of request messages whose uniform resource identifier (URI) or arguments (parameters) match a defined regular expression, use the **match request regex** command in class-map configuration mode. To remove this match criterion, use the **no** form of this command.

**match request** {**uri** | **arÅg**} **regex** *parameter-map-name*

**no match request** {**uri** | **arg**} **regex** *parameter-map-name*

## Syntax Description

| | |
|---|---|
| **uri** \| **arg** | Firewall policy will search the URI or argument as the match criterion. |
| *parameter-map-name* | HTTP-based parameter map as specified via the **parameter-map type** command. |

## Command Default

URI or parameter matching is not enabled.

## Command Modes

Class-map configuration (config-cmap)#

## Command History

| Release | Modification |
|---|---|
| 12.4(9)T | This command was introduced. |
| 15.1(1)T | The **not** keyword was added. |

## Usage Guidelines

Use the **match request uri regex** command to block custom URLs and queries; use the **match request arg regex** command to block all messages whose parameters match the configured regular inspection.

If a match is found, possible actions that can be specified within the policy are as follows: allow, reset, or log. (The log action triggers a syslog message when a match is found.)

## Examples

The following example shows how to configure an HTTP application firewall policy to block any request whose URI matches any of the following regular expressions: ".*cmd.exe," ".*money," ".*gambling".

```
parameter-map type regex uri_regex_cm
 pattern ".*cmd.exe"
 pattern ".*money"
 pattern ".*gambling"

class-map type inspect http uri_check_cm
 match request uri regex uri_regex_cm

policy-map type inspect http uri_check_pm
 class type inspect http uri_check_cm
 reset
```

The following example shows how to configure an HTTP application firewall policy to block any request whose arguments match the ".*codered" or the ".*attack" regular expressions:

```
parameter-map type regex arg_regex_cm
 pattern ".*codered"
 pattern ".*attack"

class-map type inspect http arg_check_cm
 match request arg regex arg_regex_cm

policy-map type inspect http arg_check_pm
 class type inspect http arg_check_cm
 reset
```

## Related Commands

| Command | Description |
|---|---|
| **parameter-map type** | Defines a parameter map. |
| **class-map type inspect** | Defines an inspect type class map. |
| **policy-map type inspect** | Defines an inspect type policy map. |

# match response

To configure a Session Initiation Protocol (SIP) class map to use a response method as the match criterion for permitting or denying SIP traffic, use the **match response** command in class-map configuration mode. To remove the response based match criterion from a class map, use the **no** form of this command.

**match response** {**header** *field* | **status**} **regex** *regex-parameter-map*

**no match response** {**header** *field* | **status**} **regex** *regex-parameter-map*

## Syntax Description

| | |
|---|---|
| **header** | (Optional) Identifies the SIP header field. |
| *field* | Name of the request header field. The following are valid request header fields: **accept**, **accept-encoding**, **accept-language**, **alert-info**, **allow**, **authentication-info**, **contact**, **content-disposition**, **content-encoding**, **content-language**, **content-length**, **content-type**, **error-info**, **from**, **proxy-authenticate**, **record-route**, **retry-after**, **server**, **supported**, **to**, **user-agent**, **via**, **www-authenticate**. |
| **status** | (Optional) Identifies status line in response. |
| **regex** | Indicates that a regular expression will follow. |
| *regex-parameter-map* | Name of parameter-map. |

## Command Default

No match criterion is configured.

## Command Modes

Class-map configuration (config-cmap)

## 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 this command when configuring an SIP firewall class map, after entering the **class-map type inspect** command.

## Examples

The following example shows how to specify the response method as a match criterion.

```
Router(config)# class-map type inspect sip sip-class
Router(config-cmap)# match response status regex allowed-im-users
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect sip** | Creates a class map for SIP. |

# match response body java-applet

To identify Java applets in an HTTP connection., use the **match response body java-applet** command in class-map configuration mode. To remove this inspection rule, use the **no** form of this command.

**match response body java-applet**

**no match response body java-applet**

## Syntax Description

This command has no arguments or keywords.

## Command Default

None

## Command Modes

Class-map configuration

## Command History

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

## Usage Guidelines

You can use this command when you are configuring an HTTP firewall policy map, after entering the **class-map type inspect http** command.

## Examples

The following example identifies Java applets in an HTTP connection:

```
class-map type inspect http http-class
 match response body java-applet
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect http** | Creates a class map for HTTP. |
| **ip inspect name test http java-list** | For Java applet blocking, specifies the numbered standard access list to use to determine friendly sites. |

# match response status-line regex

To specify a list of regular expressions that are to be matched against the status line of a response message, use the **match response status-line regex** command in class-map configuration mode. To remove this match criterion, use the **no** form of this command.

**match response status-line regex** *parameter-map-name*

**no match response status-line regex** *parameter-map-name*

## Syntax Description

| | |
|---|---|
| *parameter-map-name* | Name of parameter map. |

## Command Default

The status line of response messages is not considered when permitting or denying HTTP traffic.

## Command Modes

Class-map configuration

## Command History

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

## Usage Guidelines

If a match is found, possible actions that can be specified within the policy are as follows: allow, reset, or log. (The log action triggers a syslog message when a match is found.)

## Examples

The following example shows how to configure an HTTP firewall policy to log an alarm whenever an attempt is made to access a forbidden page. (A forbidden page usually contains a 403 status-code and the status line looks like “HTTP/1.0 403 page forbidden\r\n”.)

```
parameter-map type regex status_line_regex
 pattern "[Hh][Tt][Tt][Pp][/][0-9][.][0-9][ \t]+403"

class-map type inspect http status_line_cm
 match response status-line regex status_line_regex

policy-map type inspect http status_line_pm
 class type inspect http status_line_cm
 log
```

# match search-file-name

To use filenames within a search request as the match criterion, use the **match search-file-name** command in class-map configuration mode. To remove this match criterion from the configuration file, use the **no** form of this command.

**match search-file-name** [*regular-expression*]

**no match search-file-name** [*regular-expression*]

## Syntax Description

| | |
|---|---|
| *regular-expression* | (Optional) The regular expression used to identify specific filenames within a search request. For example, entering ".exe" as the regular expression would classify the filenames containing the string ".exe" as matches for the traffic policy.<br>If this argument is not issued, all filenames are classified, as appropriate. |

## Command Default

None

## Command Modes

Class-map configuration

## Command History

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

## Usage Guidelines

Use the **match search-file-name** command to configure the Cisco IOS Firewall to block filenames within a search request for clients using the eDonkey peer-to-peer (P2P) protocol.

**Note** This command is available only for the eDonkey P2P protocol.

## Examples

The following example shows how to configure a Cisco IOS Firewall to block filename searches for ".exe" and permit file transfers within the eDonkey protocol:

```
! Select eDonkey protocol requiring L7 policies
class-map type inspect match-any my-restricted-p2p
 match protocol edonkey signature
!
! Configure Edonkey to look for "*.exe" in searches
class-map type inspect edonkey my-edonkey-exe
 match search-file-name "*.exe"
!
! Configure Edonkey to look for file-transfers
class-map type inspect edonkey my-edonkey-file-tx
 match file-transfer *
!
```

```
! Configure P2P Layer 7 policy map
policy-map type inspect p2p my-p2p-policy
! class type inspect edonkey my-edonkey-exe
 reset
 class type inspect edonkey my-edonkey-file-tx
 allow
 log
!
!
```

**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. |

# match sender address regex

To specify spam e-mail from suspected domains and user accounts to be restricted, use the **match sender address regex** command in class-map configuration mode. To disable this inspection parameter, use the **no** form of this command.

**match sender address regex** *parameter-map-name*

**no match sender address regex** *parameter-map-name*

## Syntax Description

| | |
|---|---|
| *parameter-map-name* | Specifies the parameter-map name class, which is the name of a specific traffic pattern. This pattern is a Cisco IOS regular expression (regex) pattern for a class-map. |

## Command Default

The parameter-map name class is not defined.

## Command Modes

Class-map configuration

## Command History

| Release | Modification |
|---|---|
| 12.4(20)T | This command was introduced. |
| Cisco IOS XE Release 3.2S | This command was integrated into Cisco IOS XE Release 3.2S. |

## Usage Guidelines

The **match sender address regex** command helps to match the parameter-map name of a specific traffic pattern that specifies a sender domain or e-mail address in the SMTP traffic. The specified pattern is scanned in the parameter for the SMTP **MAIL FROM:** command.

## Examples

The following example shows how to configure an SMTP application firewall policy to restrict an e-mail sender from a suspected domain:

```
parameter-map type regex bad-guys
 pattern "*deals\.com"
 pattern *crazyperson*@hotmail\.com

class-map type inspect smtp match-any c1
 match sender address regex bad-guys

policy-map type inspect smtp p1
 class type inspect smtp c1
 log
 reset
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type inspect smtp** | Creates a class map for the SMTP protocol so that the match criteria is set to match criteria for this class map. |
| **parameter-map type regex** | Enters the parameter-map name of a specific traffic pattern. |
| **pattern** | Cisco IOS regular expression (regex) pattern that matches the traffic pattern for the e-mail sender or user accounts from suspected domains that are causing the spam e-mail. |

# match server-domain urlf-glob

To configure the match criteria for a local URL filtering class map on the basis of server domain name, use the **match server-domain urlf-glob** command in class-map configuration mode. To remove the domain name match criteria from a URL filtering class map, use the **no** form of this command.

**match server-domain urlf-glob** *parameter-map-name*

**no match server-domain urlf-glob** *parameter-map-name*

## Syntax Description

| | |
|---|---|
| *parameter-map-name* | Name of the parameter map. |

## Command Default

No match criteria are configured.

## Command Modes

Class-map configuration (config-cmap)

## 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 **match server-domain urlf-glob** command specifies the server domain matches for local URL filtering. Typically, you use this command in two class maps: one to specify trusted domains and one to specify untrusted domains. You must configure the **urlf-glob** keyword with the **parameter-map type urlf-glob** command and create the local filtering class with the **class-map type urlfilter** command before using this command, otherwise you will receive an error message.

## Examples

The following example shows the configuration for trusted domains and untrusted domains:

```
parameter-map type urlf-glob trusted-domain-param
 pattern www.example.com
 pattern *.example1.com

class-map type urlfilter match-any trusted-domain-class
 match server-domain urlf-glob trusted-domain-param

parameter-map type urlf-glob untrusted-domain-param
 pattern www.example3.com
 pattern www.example4.com

class-map type urlfilter match-any untrusted-domain-class
 match server-domain urlf-glob untrusted-domain-param
```

**Related Commands**

| Command | Description |
|---|---|
| **class-map type urlfilter** | Creates a class map to be used for matching packets to which a URL filtering policy applies. |
| **match url-keyword urlf-glob** | Specifies the match criteria for a local URL keyword filter. |
| **parameter-map type urlf-glob** | Specifies the per-policy parameters for local URL filtering of trusted domains, untrusted domains, and URL keywords. |

# match server-response any

To configure the match criterion for a SmartFilter (N2H2) or Websense URL filtering class map, use the **match server-response any** command in class-map configuration mode. To remove the match criterion, use the **no** form of this command.

**match server-response any**

**no match server-response any**

## Syntax Description

This command has no arguments or keywords.

## Command Default

No match criterion is configured.

## Command Modes

Class-map configuration (config-cmap)

## 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 **match server-response any** command to specify that any response from the SmartFilter or Websense server results in a match. Use this command after you have created a class map with the **class-map type urlfilter n2h2** or the **class-map type urlfilter websense** command:

## Examples

The following example shows the configuration for a SmartFilter class:

```
class-map type urlfilter n2h2 match-any smartfilter-class
 match server-response any
```

The following example shows the configuration for a Websense class:

```
class-map type urlfilter websense match-any websense-class
 match server-response any
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type urlfilter** | Creates a class map to which a URL filtering policy applies. |

# match service

To specify a match criterion for any supported Instant Messenger (IM) protocol, use the **match service** command in class-map configuration mode. To remove the match criterion from the configuration file, use the **no** form of this command.

**match service** {**any** | **text-chat**}

**no match service** {**any** | **text-chat**}

## Syntax Description

| | |
|---|---|
| **any** | Matches any type of service within the given IM protocol with the exception of text chat messages. |
| **text-chat** | Matches packets for text chat messages. |

## Command Default

None

## Command Modes

Class-map configuration (config-cmap)

## Command History

| Release | Modification |
|---|---|
| 12.4(9)T | This command was introduced. |
| 12.4(20)T | Support for I Seek You (ICQ) and Windows Messenger IM Protocols was added. |

## Usage Guidelines

Use the **match service** command to configure the Cisco IOS Firewall to create a match criterion on the basis of text chat messages or for any available service within a given IM protocol.

Before you can use the **match service** command, you must issue the **class-map type inspect** command and specify one of the following IM protocols: AOL, ICQ, MSN Messenger, Yahoo Messenger, and Windows Messenger.

## Examples

The following example shows how to configure an AOL IM policy that permits text chat and blocks any MSN IM service:

```
class-map type inspect aol match-any l7cmap-service-text-chat
 match service text-chat
!
class-map type inspect msnmsgr match-any l7cmap-service-any
 match service any

! Allow text-chat, reset if any other service, alarm for both
policy-map type inspect im l7pmap
class type inspect aol l7cmap-service-text-chat
allow
log
!
```

```
class type inspect msnmsgr l7cmap-service-any
reset
log
```

**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. |

# match text-chat

To use text chat messages as the match criterion, use the **match text-chat** command in class-map configuration mode. To remove the match criterion from the configuration file, use the **no** form of this command.

**match text-chat** [*regular-expression*]

**no match text-chat** [*regular-expression*]

## Syntax Description

| | |
|---|---|
| *regular-expression* | (Optional) The regular expression used to identify specific eDonkey text chat messages. For example, entering ".exe" as the regular expression would classify the eDonkey text chat messages containing the string ".exe" as matches for the traffic policy.<br><br>To specify that all eDonkey text chat messages be identified by the traffic class, use an asterisk (*) as the regular expression. |

## Command Default

None

## Command Modes

Class-map configuration

## Command History

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

## Usage Guidelines

Use the **match text-chat** command to configure the Cisco IOS firewall to block text chat messages between clients using the eDonkey peer-to-peer (P2P) application.

**Note** This command is available only for the eDonkey P2P protocol.

## Examples

The following example shows how to configure all text chat messages to be classified into the "my-edonkey-exe" class map:

```
class-map type inspect edonkey match-any my-edonkey-exe
 match text-chat
```

## 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. |

# match url

To specify the URL to be associated with the URL profile that configures the SDP registrar to run HTTPS, use the **match url** command in tti-registrar configuration mode. To remove this configuration, use the **no** form of this command.

**match url** *url*

**no match url** *url*

## Syntax Description

| | |
|---|---|
| *url* | Specifies the URL to be associated with the URL profile. |

## Command Default

No URL is associated with the URL profile.

## Command Modes

Tti-registrar configuration mode (tti-registrar)

## Command History

| Release | Modification |
|---|---|
| 15.1(2)T | This command was introduced. |

## Usage Guidelines

The **match url** command is required in the SDP registrar configuration, which is used to deploy Apple iPhones on a corporate network.

## Examples

The following example configures the SDP registrar to run HTTPS in order to deploy Apple iPhones on a corporate network from global configuration mode:

```
Router(config)# crypto provisioning registrar
Router(tti-registrar)# url-profile start START
Router(tti-registrar)# url-profile intro INTRO
Router(tti-registrar)# match url /sdp/intro
Router(tti-registrar)# match authentication trustpoint apple-tp
Router(tti-registrar)# match certificate cat 10
Router(tti-registrar)# mime-type application/x-apple-aspen-config
Router(tti-registrar)# template location flash:intro.mobileconfig
Router(tti-registrar)# template variable p iphone-vpn
```

## Related Commands

| Command | Description |
|---|---|
| **crypto provisioning registrar** | Configures a device to become a registrar for the SDP exchange and enters tti-registrar configuration mode. |
| **url-profile** | Specifies a URL profile that configures the SDP registrar to run HTTPS in order to deploy Apple iPhones on a corporate network. |
| **match authentication trustpoint** | Enters the trustpoint name that should be used to authenticate the peer's certificate. |

| Command | Description |
|---|---|
| **match certificate** | Enters the name of the certificate map used to authorize the peer's certificate. |
| **mime-type** | Specifies the MIME type that the SDP registrar should use to respond to a request received through the URL profile. |
| **template location** | Specifies the location of the template that the SDP Registrar should use while responding to a request received through the URL profile. |
| **template variable p** | Specifies the value that goes into the OU field of the subject name in the certificate to be issued. |

# match url category

To configure the match criteria for a Trend-Micro URL filtering class map on the basis of the specified URL category, use the **match url category** command in class-map configuration mode. To remove the URL category match criteria from a URL filtering class map, use the **no** form of this command.

**match url category** *category-name*

**no match url category** *category-name*

## Syntax Description

| | |
|---|---|
| *category-name* | Name of the URL category. |

## Command Default

No match criteria are configured.

## Command Modes

Class-map configuration (config-cmap)

## 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 **match url category** command specifies the name of the URL category to be used as the match criteria against which packets are checked to determine whether they belong to the class specified by the class map. Before you can use the **match url category** command, you must first use the **class-map type urlfilter** command to specify the name of the class whose match criteria you want to establish.

To display a list of supported URL categories, use the **match url category ?** command in class map configuration mode.

## Examples

The following example specifies a class map for Trend Micro filtering called drop-category and configures the URL categories Gambling and Personals-Dating as match criteria:

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

## Related Commands

| Command | Description |
|---|---|
| **class-map type urlfilter** | Creates a class map to be used for matching packets to which a URL filtering policy applies. |
| **match url reputation** | Specifies a match criterion for a URL filtering class map on the basis of URL reputation. |

# match url-keyword urlf-glob

To configure the match criteria for a local URL filtering class map on the basis of the URL keyword, use the **match url-keyword urlf-glob** command in class-map configuration mode. To remove the keyword match criteria from a URL filtering class map, use the **no** form of this command.

**match url-keyword urlf-glob** *parameter-map-name*

**no match url-keyword urlf-glob** *parameter-map-name*

## Syntax Description

| | |
|---|---|
| *parameter-map-name* | Name of the parameter map. |

## Command Default

No match criteria are configured.

## Command Modes

Class-map configuration (config-cmap)

## 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 **match url-keyword urlf-glob** command specifies URL keyword matches for local URL filtering. Typically, you use this command to specify the URL keywords for which you want to block access. You must configure the **urlf-glob** keyword with the **parameter-map type urlf-glob** command and create the local filtering class with the **class-map type urlfilter** command before using this command, otherwise you will receive an error message.

## Examples

The following example shows the use of:

- The **parameter-map type urlf-glob** command to configure the the keyword matching patterns.
- The **class-map type urlfilter** command to create the local URL filtering class keyword class.
- The **match url-keyword urlf-glob** command to specify the matching criteria for the class.

```
parameter-map type urlf-glob keyword-param
 pattern example
 pattern www.example1
 pattern example3

class-map type urlfilter match-any keyword-class
 match url-keyword urlf-glob keyword-param
```

**Related Commands**

| Command | Description |
|---|---|
| **class-map type urlfilter** | Creates a class map to be used for matching packets to which a URL filtering policy applies. |
| **match server-domain urlf-glob** | Specifies the match criteria for a local domain name filter. |
| **parameter-map type urlf-glob** | Specifies the per-policy parameters for local URL filtering of trusted domains, untrusted domains, and URL keywords. |

# match url reputation

To configure the match criteria for a Trend-Micro URL filtering class map on the basis of the specified URL reputation, use the **match url reputation** command in class-map configuration mode. To remove the URL reputation match criteria from a URL filtering class map, use the **no** form of this command.

**match url reputation** *reputation-name*

**no match url reputation** *reputation-name*

## Syntax Description

| | |
|---|---|
| *reputation-name* | Name of the URL reputation. |

## Command Default

No match criteria are configured.

## Command Modes

Class-map configuration (config-cmap)

## 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 **match url reputation** command specifies the name of the URL reputation to be used as a match criterion against which packets are checked to determine whether they belong to the class specified by the class map. Before you can use the **match url reputation** command, you must first use the **class-map type urlfilter** command to specify the name of the class whose match criteria you want to establish.

To display a list of supported URL reputations, use the **match url reputation ?** command in class map configuration mode.

## Examples

The following example specifies a class map for Trend Micro filtering called drop-reputation and configures the URL reputations ADWARE and PHISHING as match criteria:

```
class-map type urlfilter trend match-any drop-reputation
 match url reputation ADWARE
 match url reputation PHISHING
```

## Related Commands

| Command | Description |
|---|---|
| **class-map type urlfilter** | Creates a class map to be used for matching packets to which a URL filtering policy applies. |
| **match url category** | Specifies a match criterion for a URL filtering class map on the basis of URL category. |

# match user-group

To configure the match criterion for a class map on the basis of the specified user group, use the **match user-group** command in class-map configuration mode. To remove user-group based match criterion from a class map, use the **no** form of this command.

**match user-group** *group-name*

**no match user-group** *group-name*

## Syntax Description

| | |
|---|---|
| *group-name* | Name of the user-group used as a matching criterion. |

## Command Default

No match criterion is configured.

## Command Modes

Class-map configuration (config-cmap)

## Command History

| Release | Modification |
|---|---|
| 12.4(20)T | This command was introduced. |

## Usage Guidelines

To use the **match user-group** command, you must first enter the **class-map** command to specify the name of the class whose match criteria you want to establish.

## Examples

The following example specifies a class map called ftp and configures the user-group as a match criterion:

```
Router(config)# class-map type inspect match-all auth_proxy_ins_cm
Router(config-cmap)# description
!
Inspect Type Class-map for auth_proxy_ug
!
Router(config-cmap)# match protocol telnet
Router(config-cmap)# match user-group auth_proxy_ug
Router(config-cmap)# exit
Router(config)# class-map type inspect match-all eng_group_ins_cm
Router(config-cmap)# description
!
Inspect Type Class-map for eng_group_ug
!
Router(config-cmap)# match protocol telnet
Router(config-cmap)# match user-group eng_group_ug
Router(config-cmap)# exit
Router(config)# class-map type inspect match-all manager_group_ins_cm
Router(config-cmap)# description
!
Inspect Type Class-map for manager_group_ug
!
Router(config-cmap)# match protocol ftp
Router(config-cmap)# match user-group manager_group_ug
```

```
Router(config-cmap)# end
```

**Related Commands**

| Command | Description |
|---|---|
| **class-map** | Creates a class map to be used for matching packets to a specified class. |
| **user-group** | Defines the user-group associated with the identity policy. |

# max-destination

To configure the maximum number of destinations that a firewall can track, use the **max-destination** command in profile configuration mode. To disable the configuration, use the **no** form of this command.

**max-destination** *number*

**no max-destination** *number*

**Syntax Description**

| | |
|---|---|
| *number* | Maximum destination value. Valid values are from 1 to 4294967295. |

**Command Default**

The maximum number of destinations that a firewall can track is not configured.

**Command Modes**

Profile configuration (config-profile)

**Command History**

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

**Usage Guidelines**

You must configure the **parameter-map type inspect-zone** command before you can configure the **max-destination** command.

The firewall creates an entry for each destination to track the rate of TCP synchronization (SYN) flood packets arriving from a zone to a destination address. The number of entries that a firewall creates should be limited, so that these entries do not consume a lot of memory during a denial-of-service (DoS) attack. The **max-destination** command configures the maximum number of destinations that a firewall can track. When the maximum limit is reached, the SYN packets to a destination are dropped.

**Examples**

The following example shows how to set the maximum number of destinations that a firewall can track to 10000:

```
Router(config)# parameter-map type inspect-zone
Router(config-profile)# max-destination 10000
Router(config-profile)# end
```

**Related Commands**

| Command | Description |
|---|---|
| **parameter-map type inspect-zone** | Configures a parameter map of type inspect zone and enters profile configuration mode. |

# max-header-length

To permit or deny HTTP traffic on the basis of the message header length, use the **max-header-length** command in appfw-policy-http configuration mode. To disable this inspection parameter, use the **no** form of this command.

**max-header-length request** *bytes* **response** *bytes* **action** {**reset** | **allow**} [**alarm**]

**no max-header-length request** *bytes* **response** *bytes* **action** {**reset** | **allow**} [**alarm**]

## Syntax Description

| | |
|---|---|
| **request** *bytes* | Maximum header length, in bytes, allowed in the request message. Number of bytes range: 0 to 65535. |
| **response** *bytes* | Maximum header length, in bytes, allowed in the response message. Number of bytes range: 0 to 65535. |
| **action** | Messages that exceed the maximum size 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 issued, all traffic is permitted.

## Command Modes

appfw-policy-http configuration

## Command History

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

## Usage Guidelines

All message header lengths exceeding the configured maximum size will be subjected to the specified action (**reset** or **allow**).

## 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
!
!
```

# max-incomplete

To define the number of existing half-open sessions that will cause the Cisco IOS firewall to start and stop deleting half-open sessions, use the **max-incomplete** command in parameter-map type inspect configuration mode. To disable this function, use the **no** form of this command.

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

**no max-incomplete** {**low** *number-of-connections* | **high** *number-of-connections*}

## Syntax Description

| | |
|---|---|
| **low** *number-of-connections* | Minimum number of half-open sessions that will cause the Cisco IOS firewall to stop deleting half-open sessions. The default is unlimited. |
| **high** *number-of-connections* | Maximum number of half-sessions after which the Cisco IOS firewall will start deleting half-open sessions. The default is unlimited. |

## Command Default

The maximum number is unlimited and no half-open sessions are deleted.

## Command Modes

Parameter-map type inspect configuration

## Command History

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

## Usage Guidelines

When you are configuring an inspect type parameter map, you can enter the **max-incomplete** subcommand after you enter the **parameter-map type inspect** command.

Enter the **max-incomplete** command twice. The first command specifies a high number at which the system will start deleting half-open sessions. The second command specifies a low number at which the system will stop deleting half-open sessions.

For more detailed information about creating a parameter map, see the **parameter-map type inspect** command.

## Examples

The following example shows how to specify that the Cisco IOS firewall will stop deleting half-open sessions when there is a minimum of 800 half-open sessions and a maximum of 10000 half-open sessions:

```
parameter-map type inspect internet-policy
 max-incomplete high 10000
 max-incomplete low unlimited 800
```

**Related Commands**

| Command | Description |
|---|---|
| **ip inspect max-incomplete high** | Defines the number of existing half-open sessions that will cause the software to start deleting half-open sessions. |
| **ip inspect max-incomplete low** | Defines the number of existing half-open sessions that will cause the software to stop deleting half-open sessions. |
| **parameter-map type inspect** | Configures an inspect parameter map for connecting thresholds, timeouts, and other parameters pertaining to the **inspect** action. |

# max-logins

To limit the number of simultaneous logins for users in a specific server group, use the **max-logins** command in global configuration mode. To remove the number of connections that were set, use the **no** form of this command.

**max-logins** *number-of-users*

**no max-logins** *number-of-users*

## Syntax Description

| | |
|---|---|
| *number-of-users* | Number of logins. The value ranges from 1 through 10. |

## 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

The **crypto isakmp client configuration group** command must be configured before this command can be configured.

This command makes it possible to mimic the functionality provided by some RADIUS servers for limiting the number of simultaneous logins for users in that group.

The **max-users** and **max-logins** keywords can be enabled together or individually to control the usage of resources by any groups or individuals.

## Examples

The following example shows that the maximum number of logins for users in server group “cisco” has been set to 8:

```
Router (config)# crypto isakmp client configuration group cisco
Router (config)# max-logins 8
```

The following shows the RADIUS attribute-value (AV) pairs for the maximum users and maximum logins parameters:

```
ipsec:max-users=1000
ipsec:max-logins=1
```

## Related Commands

| Command | Description |
|---|---|
| **crypto isakmp client configuration group** | Specifies to which group a policy profile will be defined. |
| **max-users** | Limits the number of connections to a specific server group. |

# max-request

To specify the maximum number of outstanding requests that can exist at any given time, use the **max-request** command in URL parameter-map configuration mode. To disable this feature, use the **no** form of this command.

**max-request** *number-of-requests*

**no max-request** *number-of-requests*

## Syntax Description

| | |
|---|---|
| *number-of-requests* | Maximum number of pending requests that can be queued to the urlfiltering server. |

## Command Default

None

## Command Modes

URL parameter-map configuration

## Command History

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

## Usage Guidelines

When you are creating or modifying a URL parameter map, you can enter the **max-request** subcommand after you enter the **parameter-map type urlfilter** command. For more detailed information about creating a parameter map, see the **parameter-map type urlfilter** command.

## Examples

The following example specifies that there can be a maximum of 80 outstanding requests at a given time:

```
parameter-map type urlfilter u1
 max-request 80
```

## Related Commands

| Command | Description |
|---|---|
| **parameter-map type urlfilter** | Creates or modifies a parameter map for URL filtering parameters. |

# max-resp-pak

To specify the maximum number of HTTP responses that the Cisco IOS firewall can keep in its packet buffer, use the **max-resp-pak** command in URL parameter-map configuration mode. To disable this feature, use the **no** form of this command.

**max-resp-pak** *number-of-responses*

**no max-resp-pak** *number-of-responses*

## Syntax Description

| | |
|---|---|
| *number-of-responses* | Maximum number of HTTP responses that the firewall can keep in its packet buffer before it starts dropping responses. |

## Command Default

None

## Command Modes

URL parameter-map configuration

## Command History

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

## Usage Guidelines

When you are creating or modifying a URL parameter map, you can enter the **max-resp-pak** subcommand after you enter the **parameter-map type urlfilter** command. For more detailed information about creating a parameter map, see the **parameter-map type urlfilter** command.

## Examples

The following example specifies that there can be a maximum of 200 HTTP responses in the packet buffer:

```
parameter-map type urlfilter eng-filter-profile
 max-resp-pak 200
```

## Related Commands

| Command | Description |
|---|---|
| **parameter-map type urlfilter** | Creates or modifies a parameter map for URL filtering parameters. |

# max-retry-attempts

To set the maximum number of retries before Single SignOn (SSO) authentication fails, use the **max-retry-attempts** command in webvpn sso server configuration mode. To remove the number of retries that were set, use the **no** form of this command.

**max-retry-attempts** *number-of-retries*

**no max-retry-attempts** *number-of-retries*

## Syntax Description

| | |
|---|---|
| *number-of-retries* | Number of retries. Value = 1 through 5. Default = 3. |

## Command Default

A maximum number of retries is not set. If this command is not configured, the default is 3 retries.

## Command Modes

Webvpn sso server configuration

## Command History

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

## Usage Guidelines

This command is useful for networks that are congested and tend to have losses. Corporate networks are generally not affected by congestion or losses.

## Examples

The following example shows that the maximum number of retries is 3:

```
webvpn context context1
 sso-server test-sso-server
  max-retry-attempts 3
```

## Related Commands

| Command | Description |
|---|---|
| **webvpn context** | Enters webvpn context configuration mode to configure the SSL VPN context. |

# max-uri-length

To permit or deny HTTP traffic on the basis of the uniform resource identifier (URI) length in the request message, use the **max-uri-length** command in appfw-policy-http configuration mode. To disable this inspection parameter, use the **no** form of this command.

**max-uri-length** *bytes* **action** {**reset** | **allow**} [**alarm**]

**no max-uri-length** *bytes* **action** {**reset** | **allow**} [**alarm**]

## Syntax Description

| | |
|---|---|
| *bytes* | Number of bytes ranging from 0 to 65535. |
| **action** | Messages that exceed the maximum URI length 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 issued, all traffic is permitted.

## Command Modes

appfw-policy-http configuration

## Command History

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

## Usage Guidelines

All URI lengths exceeding the configured value will be subjected to the specified action (**reset** or **allow**).

## 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
!
!
```

# max-users

To limit the number of connections to a specific server group, use the **max-users** command in global configuration mode. To remove the number of connections that were set, use the **no** form of this command.

**max-users** *number-of-users*

**no max-users** *number-of-users*

## Syntax Description

| | |
|---|---|
| *number-of-users* | Number of users. The value ranges from 1 through 5000. |

## Command Modes

Global configuration (config)

## Command History

| Release | Modification |
|---|---|
| 12.2(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

The **crypto isakmp client configuration group** command must be configured before this command can be configured.

This command makes it possible to mimic the functionality provided by some RADIUS servers for limiting the number of connections to a specific server group.

The **max-users** and **max-logins** keywords can be enabled together or individually to control the usage of resources by any groups or individuals.

## Examples

The following example shows that the maximum number of connections to server group “cisco” has been set to 1200:

```
Router (config)# crypto isakmp client configuration group cisco
Router (config)# max-users 1200
```

The following shows the RADIUS attribute-value (AV) pairs for the maximum users and maximum logins parameters:

```
ipsec:max-users=1000
ipsec:max-logins=1
```

**Related Commands**

| Command | Description |
|---|---|
| **crypto isakmp client configuration group** | Specifies to which group a policy profile will be defined. |
| **max-logins** | Limits the number of simultaneous logins for users in a specific server group. |

# max-users (WebVPN)

To limit the number of connections to an SSL VPN that will be permitted, use the **max-users** command in webvpn context configuration mode. To remove the connection limit from the SSL VPN context configuration, use the **no** form of this command.

**max-users** *number*

**no max-users**

## Syntax Description

| | |
|---|---|
| *number* | Maximum number of SSL VPN user connections. A number from 1 to 1000 can be entered for this argument. |

## Command Default

The following is the default if this command is not configured or if the **no** form is entered:

*number* : 1000

## Command Modes

Webvpn context configuration

## Command History

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

## Examples

The following example configures a limit of 500 user connections that will be accepted by the SSL VPN:

```
Router(config)# webvpn context context1
Router(config-webvpn-context)# max-users 500
```

## Related Commands

| Command | Description |
|---|---|
| **webvpn context** | Enters webvpn context configuration mode to configure the SSL VPN context. |

# mime-type

To specify the Multipurpose Internet Mail Extensions (MIME) type that the SDP registrar should use to respond to a request received through the URL profile, use the **mime-type** command in tti-registrar configuration mode. To remove this configuration, use the **no** form of this command.

**mime-type** *mime-type*

**no mime-type** *mime-type*

## Syntax Description

| | |
|---|---|
| *mime-type* | Specifies the MIME type. |

## Command Default

No MIME type is configured for the SDP registrar.

## Command Modes

Tti-registrar configuration mode (tti-registrar)

## Command History

| Release | Modification |
|---|---|
| 15.1(2)T | This command was introduced. |

## Usage Guidelines

The **mime-type** command is required in the SDP registrar configuration, which is used to deploy Apple iPhones on a corporate network.

## Examples

The following example configures the SDP registrar to run HTTPS in order to deploy Apple iPhones on a corporate network from global configuration mode:

```
Router(config)# crypto provisioning registrar
Router(tti-registrar)# url-profile start START
Router(tti-registrar)# url-profile intro INTRO
Router(tti-registrar)# match url /sdp/intro
Router(tti-registrar)# match authentication trustpoint apple-tp
Router(tti-registrar)# match certificate cat 10
Router(tti-registrar)# mime-type application/x-apple-aspen-config
Router(tti-registrar)# template location flash:intro.mobileconfig
Router(tti-registrar)# template variable p iphone-vpn
```

## Related Commands

| Command | Description |
|---|---|
| **crypto provisioning registrar** | Configures a device to become a registrar for the SDP exchange and enters tti-registrar configuration mode. |
| **url-profile** | Specifies a URL profile that configures the SDP registrar to run HTTPS in order to deploy Apple iPhones on a corporate network. |
| **match authentication trustpoint** | Enters the trustpoint name that should be used to authenticate the peer's certificate. |

| Command | Description |
|---|---|
| **match certificate** | Enters the name of the certificate map used to authorize the peer's certificate. |
| **match url** | Specifies the URL to be associated with the URL profile. |
| **template location** | Specifies the location of the template that the SDP Registrar should use while responding to a request received through the URL profile. |
| **template variable p** | Specifies the value that goes into the OU field of the subject name in the certificate to be issued. |