# Tag and Template

**First Published: February 27, 2006**
**Last Updated: February 27, 2006**

The Tag and Template feature allows network administrators to define enforcement policies on a local device and have a RADIUS server specify the policy selector to be enforced. This feature can be applied to a Network Admission Control (NAC) architecture.

### Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Tag and Template" section on page 33.

### Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

# Contents

- Prerequisites for Tag and Template, page 2
- Requirements for Tag and Template, page 2
- Information About Tag and Template, page 2
- How to Configure Tag and Template, page 3
- Configuration Examples for Tag and Template, page 8
- Where to Go Next, page 9
- Additional References, page 9
- Command Reference, page 10
- Feature Information for Tag and Template, page 33

CISCO SYSTEMS

**Corporate Headquarters:**
**Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA**

# Prerequisites for Tag and Template

- You must have a Cisco IOS image that supports the Modular Quality of Service (QoS) command-line interface (CLI).

# Requirements for Tag and Template

- To apply the enforcement policies, the identity policy and access groups that are associated with the identity policy have to be configured for Tag and Template.

# Information About Tag and Template

Before configuring Tag and Template, you should understand the following concepts:

- Tag and Template Overview, page 2

## Tag and Template Overview

In a typical Network Admission Control deployment, an access control server (ACS) or a RADIUS server is used for validating the user posture information and for applying the policies on the network access device (NAD). A centralized ACS can be used to support multiple NADs. This solution has inherent problems associated with it, namely:

- Version control of policies.Typically, a specific NAD that is running a Cisco IOS image may support some ACLs, and another NAD may support a different version. Managing different versions can be a problem.
- Users connect on different interfaces to the NAD, and on the basis of the interface type, the policies that can be applied to the user can change, and the NAD can determine the policies to be applied. In the current architecture, the ACS sends the same set of policies to all the NADs when a profile is matched, which does not give enough control to the administrator to configure the polices on the basis of the NAD configuration.

To overcome the above problems, the Tag and Template concept has been introduced. The concept is that the ACS maps users to specific groups and associates a tag with them. For example, the Usergroup1 user group may have a tag with the name “usergroup1.” When the NAD queries the ACS for the policies, the ACS can return the tag that is associated with the user group. When this tag is received at the NAD, the NAD can map the tag to a specific template that can have a set of policies that are associated with the user group. This mapping provides administrators with the flexibility to configure the template on a NAD basis, and the policies can change from NAD to NAD even though the tag is the same.

In summary, a template must be configured on the NAD, and the template must be associated with a tag. When the ACS sends the policies back to the NAD, the template that matches the tag that was received from the ACS is used.

# How to Configure Tag and Template

This section includes the following procedures:


- Defining a Class Map for a Specific Type and Associating Match Conditions with It, page 3
- Associating the Class Map with the Policy Map and Applying Actions for Classes That Match, page 4
- Associating the Service Policy with a Specific IP Admission Rule, page 5
- Monitoring the Template Configuration, page 6
- Verifying the Template Configuration, page 7


## Defining a Class Map for a Specific Type and Associating Match Conditions with It

To define a class map and associate match conditions with it, perform the following steps.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type tag [match-all | match-any]** *class-map-name*
4. **match port-type** {**routed** | **switched**}
5. **match tag** *tag-name*

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **class-map type tag [match-all \| match-any]** *class-map-name*<br><br>**Example:**<br>`Router (config)# class-map tag match-all group1_class` | Creates a class map to be used for matching packets to a specified class and enters class-map configuration mode. |
| **Step 4** | **match port-type** {**routed** \| **switched**}<br><br>**Example:**<br>`Router (config-cmap)# match port-type routed` | Matches the access policy on the basis of the port for a class map. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | **match tag** *tag-name*<br><br>**Example:**<br>`Router (config-cmap)# match tag group1_class` | Specifies the tag to be matched for a tag type of class map. |

## What to Do Next

Associate the class map with the policy map and apply actions for classes that match.

# Associating the Class Map with the Policy Map and Applying Actions for Classes That Match

To associate the class map with the policy map and apply actions for classes that match, perform the following steps.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type control tag** *policy-map-name*
4. **class type tag** {*class-name*} [**insert-before** {*class-name*}]
5. **identity policy** *policy-name*

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **policy-map type control tag** *policy-map-name*<br><br>**Example:**<br>`Router (config)# policy-map type control tag usergroup1_pmap` | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy and enters policy-map configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | **class type tag** {*class-name*} [**insert-before** {*class-name*}]<br><br>**Example:**<br>`Router (config-pmap)# class type tag usergroup1_class` | Associates a class map with a policy map. |
| **Step 5** | **identity policy** *policy-name*<br><br>**Example:**<br>`Router (config-pmap)# identity policy usergroup1_iden_policy` | Associates an identity policy with the class map. |

## What to Do Next

Associate the service policy with a specific IP admission table.

# Associating the Service Policy with a Specific IP Admission Rule

The policy map defined above can be associated with an IP authentication proxy or IP admission rule. To associate the map with the IP authentication proxy or IP admission rule, perform the following steps.

**Note** There can be multiple policy maps, and each one can be associated with a different IP admission rule even though an IP admission rule can have only one instance of the policy map.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip admission name** *admission-name* [**eapoudp** | **proxy** {**ftp** | **http** | **telnet**} | **service-policy type tag** {*service-policy-name*} ] [**list** {*acl* | *acl-name*}]

   or

   **ip auth-proxy name** *auth-proxy-name* {**ftp** | **http** | **telnet**}[**inactivity-timer** *min*] [**absolute-timer** *min*] [**list** {*acl* | *acl-name*}] [**service-policy type tag** {*service-policy-name*} ]

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **ip admission name** *admission-name* [**eapoudp** \| **proxy** {**ftp** \| **http** \| **telnet**} \| **service-policy type tag** {*service-policy-name*} ] [**list** {*acl* \| *acl-name*}]<br><br>or<br><br>**ip auth-proxy name** *auth-proxy-name* {**ftp** \| **http** \| **telnet**}[**inactivity-timer** *min*] [**absolute-timer** *min*] [**list** {*acl* \| *acl-name*}] [**service-policy type tag** {*service-policy-name*} ]<br><br>**Example:**<br>`Router (config)# ip admission name nac eapoudp service-policy type tag usergroup1_iden_policy`<br><br>or<br><br>`Router (config)# ip auth-proxy name nac eapoudp service-policy type tag usergroup1_iden_policy` | Associates the policy map with an IP network admission control rule.<br>• The service policy name must be the same as the policy map name.<br>or<br>Associates the policy map with an authentication proxy rule. |

# Monitoring the Template Configuration

To monitor the template configuration, perform the following steps.

### SUMMARY STEPS

1. **enable**
2. **debug tag-template event**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| Step 2 | **debug tag-template event**<br><br>**Example:**<br>`Router# debug tag-template event` | Displays the tag application on a session (an Authentication Proxy or Extensible Authentication Protocol over UDP [EAPoUDP] session). |

# Verifying the Template Configuration

To verify the template configuration, perform the following steps. The **show** commands can be used individually or together.

### SUMMARY STEPS

1. **enable**
2. **show class-map type tag** *class-map-name*
3. **show epm session ip** {*ip-address* | **summary**}
4. **show policy-map type control tag** *type-name*

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| Step 1 | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| Step 2 | **show class-map type tag** *class-map-name*<br><br>**Example:**<br>`Router# show class-map type tag map1` | Displays all class maps and their matching criteria. |
| Step 3 | **show epm session ip** {*ip-address* \| **summary**}<br><br>**Example:**<br>`Router# show epm session ip 10.1.1.1` | Displays whether tag policies or authentication, authorization, and accounting (AAA) policies are actually applied to a service policy application. |
| Step 4 | **show policy-map type control tag** *type-name*<br><br>**Example:**<br>`Router# show policy-map type control tag type1` | Displays a template configuration when applying access policies on Layer 2 and Layer 3 interfaces. |

# Configuration Examples for Tag and Template

This section provides the following configuration example.

- Typical Tag and Template Configuration: Example, page 8

## Typical Tag and Template Configuration: Example

In the following service policy (Tag and Template) example, tags named “healthy” and “non_healthy” can be received from an AAA server, the policy map is defined on the NAD, and the tag policy type is associated with the IP admission name “greentree.”

### Class Map Definition for the “healthy class” Type Tag

```
Router (config)# class-map type tag healthy_class
Router(config-cmap)# match tag healthy
Router(config-cmap)# end
```

### Class Map Definition for the “non_healthy_class” Type Tag

```
Router (config)# class-map type tag non_healthy_class
Router (config-cmap)# match tag non_healthy
Router (config-cmap)# end
```

### Policy Map Is Defined

```
! The following line will be associated with the IP admission name.
Router (config)# policy-map type control tag global_class
! The following line refers to the class map that was defined above.
Router (config-pmap)# class healthy_class
Router (config-pmap-c)# identity policy healthy_policy
Router(config-pmap-c)# exit
The following line refers to the non_healthy class that was defined above.
Router (config-pmap)# class non_healthy_class
Router(config-pmap-c)# identity policy non_healthy_policy
Router (config-pmap-c)# end
```

### Identity Policy Can Be Defined As Follows

```
Router (config)# identity policy healthy_policy
! The following line is the IP access list for healthy users.
Router (config-identity-policy)# access-group healthy
Router (config-identity-policy)# end
Router (config)# identity policy non_healthy_policy
Router (config-identity-policy)# access-group non_healthy
Router (config-identity-policy)# end
```

### Access Lists Can Be Defined As Follows

```
Router (config)# ip access-list extended healthy_class
! The following line can be anything, but as an example, traffic is being allowed.
Router (config-ext-nacl)# permit ip any any
Router (config-ext-nac)# end
Router (config)# ip access-list extended non_healthy_class
! The following line is only an example. In practical cases, you could prevent a user from
accessing specific networks.
Router (config-ext-nacl)# deny ip any any
Router (config-ext-nac)# end
```

#### Policy Map That Was Defined Above Is Associated with the IP Admission Name

```
Router (config)# ip admission name greentree service-policy type tag global_class
! In the next line, the admission name can be associated with the interface.
Router (config)# interface fastethernet 1/0
Router (config-if)# ip admission greentree
```

In the above configuration, if the AAA server sends a tag named "healthy" or "non_healthy" for any host, the policies that are associated with the appropriate identity policy will be applied on the host.

# Where to Go Next

The tag attribute must be configured in the RADIUS profile using the following Cisco attribute-value (AV) pair: tag-name={tag string}.

For information about configuring RADIUS AV pairs, see the subsection “Configuring Cisco AV Pairs” in the section “Related Documents.”

# Additional References

The following sections provide references related to Tag and Template.

## Related Documents

| Related Topic | Document Title |
|---|---|
| Cisco IOS commands | *Cisco IOS Master Command List*, Release 12.4T |
| Configuring Cisco RADIUS AV pairs | The section “Configuring RADIUS” in the *Cisco IOS Security Configuration Guide*, Release 12.4 |

## Standards

| Standard | Title |
|---|---|
| No new or modified standards are supported by this feature. | — |

## MIBs

| MIB | MIBs Link |
|---|---|
| No new or modified MIBs are supported by this feature. | To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:<br>http://www.cisco.com/go/mibs |

## RFCs

| RFC | Title |
|---|---|
| No new or modified RFCs are supported by this feature. | — |

## Technical Assistance

| Description | Link |
|---|---|
| The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content. | http://www.cisco.com/techsupport |

# Command Reference

This section documents new and modified commands only.

### New Commands

- class type tag
- debug tag-template event
- identity policy (policy-map)
- match port-type
- match tag (class-map)
- show epm session ip

### Modified Commands

- class-map
- ip admission name
- ip auth-proxy name
- policy-map
- show class-map
- show policy-map

# class-map

To create a class map to be used for matching packets to a specified class, use the **class-map** command in global configuration mode. To remove an existing class map from the router, use the **no** form of this command.

**class-map** [**type** {**stack** | **access-control** | **port-filter** | **queue-threshold** | **tag**}]
[**match-all** | **match-any**] *class-map-name*

**no class-map** [**type** {**stack** | **access-control** | **port-filter** | **queue-threshold** | **tag**}]
[**match-all** | **match-any**] *class-map-name*

## Syntax Description

| | |
|---|---|
| **type stack** | (Optional) Enables the flexible packet matching (FPM) functionality to determine the correct protocol stack in which to examine.<br>If the appropriate protocol header description files (PHDFs) have been loaded onto the router (via the **load protocol** command), a stack of protocol headers can be defined so the filter can determine which headers are present and in what order. |
| **type access-control** | (Optional) Determines the exact pattern to look for in the protocol stack of interest.<br>**Note** You must specify a stack class map (via the **type stack** keywords) before you can specify an access-control class map (via the **type access-control** keywords). |
| **type port-filter** | (Optional) Creates a port-filter class-map that enables the TCP/UDP port policing of control plane packets.<br>When enabled it provides filtering of traffic destined to specific ports on the Control Plane host subinterface. |
| **type queue-threshold** | (Optional) Enables queue thresholding that limits the total number of packets for a specified protocol that is allowed in the control plane IP input queue. This feature applies only to control plane host subinterface. |
| **type tag** | (Optional) Creates the tag type class map that can be used to apply the access control policies on the network access device (NAD) on the basis of the tag that is received from the access control server (ACS). |
| **match-all** \| **match-any** | (Optional) Determines how packets are evaluated when multiple match criteria exist. Packets must either meet all of the match criteria (**match-all**) or one of the match criteria (**match-any**) in order to be considered a member of the class. |
| *class-map-name* | Name of the class for the class map. The name can be a maximum of 40 alphanumeric characters. The class name is used for both the class map and to configure policy for the class in the policy map. |

## Defaults

No default behavior or values

## Command Modes

Global 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. |
| 12.1(1)E | This command was integrated into Cisco IOS Release 12.1(1)E. |
| 12.4(4)T | The **type**, **stack**, and **access-control** keywords were added to support FPM.<br>The **type, port-filter** and **queue-threshold** keywords were added to support Control Plane Protection. |
| 12.4(6)T | The **tag** keyword was added. |

## Usage Guidelines

Use this command to specify the name of the class for which you want to create or modify class-map match criteria. Use of the **class-map** command enables class-map configuration mode in which you can enter one of the match commands to configure the match criteria for this class. Packets arriving at either the input or output interface (determined by how the **service-policy** command is configured) are checked against the match criteria configured for a class map to determine if the packet belongs to that class.

When configuring a class map, you can use one or more **match** commands to specify match criteria. For example, you can use the **match access-group** command, the **match protocol** command, or the **match input-interface** command. The **match** commands vary according to the Cisco IOS release. For more information about match criteria and **match** commands, see the "Modular Quality of Service Command-Line Interface (CLI)" chapter of the *Cisco IOS Quality of Service Solutions Configuration Guide*.

## Examples

The following example specifies class101 as the name of a class, and it defines a class map for this class. The class called class101 specifies policy for traffic that matches access control list 101.

```
class-map class101
  match access-group 101
```

The following example shows how to define FPM traffic classes for slammer and UDP packets. The match criteria defined within the class maps is for slammer and UDP packets with an IP length not to exceed 404 bytes, UDP port 1434, and pattern 0x4011010 at 224 bytes from start of IP header.

```
load protocol disk2:ip.phdf
load protocol disk2:udp.phdf

class-map type stack match-all ip-udp
 description "match UDP over IP packets"
 match field ip protocol eq 0x11 next udp

class-map type access-control match-all slammer
 description "match on slammer packets"
 match field udp dest-port eq 0x59A
 match field ip length eq 0x194
 match start l3-start offset 224 size 4 eq 0x4011010
```

The following example shows how to configure a port-filter policy to drop all traffic destined to closed or "nonlistened" ports except SNMP.

```
Router (config)# class-map type port-filter pf-class
Router (config-cmap)# match not port udp 123
Router (config-cmap)# match closed-ports
```

```
Router (config-cmap)# exit
Router (config)# policy-map type port-filter pf-policy
Router (config-pmap)# class pf-class
Router (config-pmap-c)# drop
Router(config-pmap-c)# end
```

The following example shows how to configure a class map for the type tag “healthy_class” and how to attach the class map “healthy_class” to the policy map.

```
Router (config)# class-map type tag healthy_class
Router(config-cmap)# match tag healthy
Router(config-cmap)# end
.
.
.
! The following line will be associated with the IP admission name.
Router (config)# policy-map type control tag global_class
! The following line refers to the class map that was defined above.
Router (config-pmap)# class healthy_class
Router (config-pmap-c)# identity policy healthy_policy
Router(config-pmap-c)# end
```

## Related Commands

| Command | Description |
|---|---|
| **class (policy-map)** | Specifies the name of the class whose policy you want to create or change, and the default class (commonly known as the class-default class) before you configure its policy. |
| **class class-default** | Specifies the default class for a service policy map. |
| **match (class-map)** | Configures the match critera for class map of the basis of port filter and/or protocol queue policies. |
| **match access-group** | Configures the match criteria for a class map on the basis of the specified ACL. |
| **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. |
| **policy-map** | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |
| **service-policy** | Attaches a policy map to an input interface or virtual circuit (VC), or an output interface or VC, to be used as the service policy for that interface or VC. |

# class type tag

To associate a class map with a policy map, use the **class type tag** command in policy map configuration mode. To disassociate the command, use the **no** form of this command.

**class type tag** {*class-name*} [**insert-before** {*class-name*}]

**no class type tag** {*class-name*} [**insert-before** {*class-name*}]

## Syntax Description

| | |
|---|---|
| *class-name* | Name of the class map. |
| **insert-before** *class-name* | (Optional) Adds a class map between any two existing class maps.<br>Inserting a new class map between two existing class maps provides more flexibility when modifying existing policy map configurations. Without this option, the class map is appended to the end of the policy map. |

## Command Default

A class map is not associated with a policy map.

## Command Modes

Policy map configuration

## Command History

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

## Usage Guidelines

If this command is used and the class is not configured, an error is generated to the user. The error may be something such as "% class map {*name*} not configured." If the class needs to be inserted before a specific class map, the **insert-before** keyword can be used. The **insert-before** keyword is typically needed if the administrator is configuring any per-host class maps and would like it inserted before a specific class map. The **class type tag** command creates the policy map class configuration mode. There can be multiple classes under the policy map.

## Examples

The following example shows the class map "usergroup1_class" is to be associated with a policy map:

```
class type tag usergroup1_class
```

## Related Commands

| Command | Description |
|---|---|
| **policy-map** | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |

# debug tag-template event

To display the tag application on a session (an Authentication Proxy or Extensible Authentication Protocol over UDP session), use the **debug tag-template event** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

**debug tag-template event**

**no debug tag-template event**

## Syntax Description

This command has no arguments or keywords.

## Command Default

Debugging is turned off.

## Command Modes

Privileged EXEC

## Command History

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

## Examples

The following example shows that the tag application on a session is to be displayed:

```
Router# debug tag-template event
```

## Related Commands

| Command | Description |
|---|---|
| **show epm sessions ip** | Displays whether tag policies have been applied. |

# identity policy (policy-map)

To create an identity policy, use the command in policy map class configuration mode. To remove the policy, use the **no** form of this command.

**identity policy** {*policy-name*}

**no identity policy** {*policy-name*}

## Syntax Description

| | |
|---|---|
| *policy-name* | Name of the policy. |

## Command Default

An identity policy is not created.

## Command Modes

Policy map configuration

## Command History

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

## Usage Guidelines

This command refers to the global identity policy that is configured on the device that contains the access policies that are to be applied. Only a single identity policy can be configured under the policy class configuration submode. If the identity policy is not defined on the device, an error is generated during the application of the policy.

## Examples

The following example shows that an identity policy is being configured:

```
Router (config)# policy-map type control tag healthy_pmap
Router (config-pmap)# class healthy_class
Router (config-pmap-class)# identity policy healthy_identity
Router (config-pmap-class)# end
```

In the following example, an identity policy named “healthy_policy” is being configured:

```
Router (config)# identity policy healthy_identity
Router (config-identity-policy)# access-group healthy_acl
Router (config-identity-policy)# end
```

## Related Commands

| Command | Description |
|---|---|
| **class type tag** | Associates a class map with a policy map. |
| **policy-map** | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |

# ip admission name

To create an IP network admission control rule, use the **ip admission name** command in global configuration mode. To remove the network admission control rule, use the **no** form of this command.

**ip admission name** *admission-name* [**eapoudp** [**bypass**] | **proxy** {**ftp** | **http** | **telnet**} | **service-policy type tag** {service-*policy-name*}] [**list** {*acl* | *acl-name*}]

**no ip admission name** *admission-name* [**eapoudp** [**bypass**] | **proxy** {**ftp** | **http** | **telnet**} | **service-policy type tag** {service-*policy-name*}] [**list** {*acl* | *acl-name*}]

## Syntax Description

| | |
|---|---|
| *admission-name* | Name of network admission control rule. |
| **eapoudp** | (Optional) Specifies IP network admission control using EAPoUDP. |
| **bypass** | (Optional) Admission rule bypasses Extensible Authentication Protocol over UDP (EAPoUDP) communication. |
| **proxy** | (Optional) Specifies authentication proxy. |
| **ftp** | Specifies that FTP is to be used to trigger the authentication proxy. |
| **http** | Specifies that HTTP is to be used to trigger authentication proxy. |
| **telnet** | Specified that Telnet is to be used to trigger authentication proxy. |
| **service-policy type tag** | (Optional) A control plane service policy is to be configured. |
| *service-policy-name* | (Optional) Control plane tag service policy that is configured using the **policy-map type control tag** {*policy name*} command, keyword, and argument. This policy map is used to apply the actions on the host when a tag is received. |
| **list** | (Optional) Associates the named rule with an access control list (ACL). |
| *acl* | Applies a standard, extended list to a named admission control rule. The value ranges from 1 through 199. |
| *acl-name* | Applies a named access list to a named admission control rule. |

## Defaults

An IP network admission control rule is not created.

## Command Modes

Global configuration

## Command History

| Release | Modification |
|---|---|
| 12.3(8)T | This command was introduced. |
| 12.4(6)T | The **bypass** and **service-policy type tag** keywords and *bypass-name* and *service-policy-name* arguments were added. |

## Usage Guidelines

The admission rule defines how you apply admission control.

You can associate the named rule with an ACL, providing control over which hosts use the admission control feature. If no standard access list is defined, the named admission rule intercepts IP traffic from all hosts whose connection-initiating packets are received at the configured interface.

The **list** keyword option allows you to apply a standard, extended (1 through 199) or named access list to a named admission control rule. IP connections that are initiated by hosts in the access list are intercepted by the admission control feature.

The **bypass** keyword allows an administrator the choice of not having to use the EAPoUDP-based posture validation for the hosts that are trying to connect on the port. The bypass can be used if an administrator knows that the hosts that are connected on the port do no have the Cisco Trust Agent client installed.

The **service policy type tag** {*service-policy-name*} keywords and argument allow you to associate the service policy of the type tag with the IP admission rule. On the network access device (NAD), a set of policies can be associated with an arbitrary tag string, and if the AAA server sends the same tag in response to the posture validation or authentication response, the policies that are associated with the tag can be applied on the host. The **service policy** keyword is an optional keyword, and if the service policy is not associated with the IP admission name, the policies that are received from the AAA server are applied on the host.

## Examples

The following example shows that an IP admission control rule is named “greentree” and that it is associated with ACL “101.” Any IP traffic that is destined to a previously configured network (using the **access-list** command) will be subjected to antivirus state validation using EAPoUDP.

```
Router (config)# ip admission name greentree eapoudp list 101
```

The following example shows that EAPoUDP bypass has been configured:

```
Router (config)# ip admission name greentree eapoudp bypass list 101
```

In the following service policy example, tags named “healthy” and “non_healthy” can be received from an AAA server, the policy map is defined on the NAD, and the tag policy type is associated with the IP admission name “greentree.”

### Class Map Definition for the “healthy class” Type Tag

```
Router (config)# class-map type tag healthy_class
Router(config-cmap)# match tag healthy
Router(config-cmap)# end
```

### Class Map Definition for the “non_healthy_class” Type Tag

```
Router (config)# class-map type tag non_healthy_class
Router (config-cmap)# match tag non_healthy
Router (config-cmap)# end
```

### Policy Map Is Defined

```
! The following line will be associated with the IP admission name.
Router (config)# policy-map type control tag global_class
! The following line refers to the class map that was defined above.
Router (config-pmap)# class healthy_class
Router (config-pmap-c)# identity policy healthy_policy
Router(config-pmap-c)# exit
The following line refers to the non_healthy class that was defined above.
Router (config-pmap)# class non_healthy_class
Router(config-pmap-c)# identity policy non_healthy_policy
Router (config-pmap-c)# end
```

### Identity Policy Can Be Defined As Follows

```
Router (config)# identity policy healthy_policy
! The following line is the IP access list for healthy users.
Router (config-identity-policy)# access-group healthy
Router (config-identity-policy)# end
Router (config)# identity policy non_healthy_policy
Router (config-identity-policy)# access-group non_healthy
Router (config-identity-policy)# end
```

### Access Lists Can Be Defined As Follows

```
Router (config)# ip access-list extended healthy_class
! The following line can be anything, but as an example, traffic is being allowed.
Router (config-ext-nacl)# permit ip any any
Router (config-ext-nac)# end
Router (config)# ip access-list extended non_healthy_class
! The following line is only an example. In practical cases, you could prevent a user from
accessing specific networks.
Router (config-ext-nacl)# deny ip any any
Router (config-ext-nac)# end
```

### Policy Map That Was Defined Above Is Associated with the IP admission name

```
Router (config)# ip admission name greentree service-policy type tag global_class
! In the next line, the admission name can be associated with the interface.
Router (config)# interface fastethernet 1/0
Router (config-if)# ip admission greentree
```

In the above configuration, if the AAA server sends a tag named "healthy" or "non_healthy" for any host, the policies that are associated with the appropriate identity policy will be applied on the host.

## Related Commands

| Command | Description |
|---|---|
| **ip address** | Sets a primary or secondary IP address for an interface. |

# ip auth-proxy name

To create an authentication proxy rule, use the **ip auth-proxy name** command in global configuration mode. To remove the authentication proxy rules, use the **no** form of this command.

**ip auth-proxy name** *auth-proxy-name* {**ftp** | **http** | **telnet**} [**inactivity-timer** *min*] [**absolute-timer** *min*] [**list** {*acl* | *acl-name*}] [**service-policy type tag {***service-policy-name***}]**

**no ip auth-proxy name** *auth-proxy-name*

## Syntax Description

| | |
|---|---|
| *auth-proxy-name* | Associates a name with an authentication proxy rule. Enter a name of up to 16 alphanumeric characters. |
| **ftp** | Specifies FTP to trigger the authentication proxy. |
| **http** | Specifies HTTP to trigger the authentication proxy. |
| **telnet** | Specifies Telnet to trigger the authentication proxy. |
| **inactivity-timer** *min* | (Optional) Overrides the global authentication proxy cache timer for a specific authentication proxy name, offering more control over timeout values. Enter a value in the range 1 to 2,147,483,647. The default value is equal to the value set with the **ip auth-proxy** command.<br>**Note** This option deprecates the **auth-cache-time** *min* option. |
| **absolute-timer** *min* | (Optional) Specifies a window in which the authentication proxy on the enabled interface is active. Enter a value in the range 1 to 65,535 minutes (45 and a half days). The default value is 0 minutes. |
| **list** {*acl* \| *acl-name*} | (Optional) Specifies a standard (1–99), extended (1–199), or named IP access list to use with the authentication proxy. With this option, the authentication proxy is applied only to those hosts in the access list. If no list is specified, all connections initiating HTTP, FTP, or Telnet traffic arriving at the interface are subject to authentication. |
| **service-policy type tag** | (Optional) A control plane service policyis to be configured. |
| *service-policy-name* | (Optional) Control plane tag service policy that is configured using the **policy-map type control tag** {*policy name*} command, keyword, and argument. This policy map is used to apply the actions on the host when a tag is received. |

## Defaults

The default value is equal to the value set with the **ip auth-proxy auth-cache-time** command.

## Command Modes

Global configuration

## Command History

| Release | Modification |
|---|---|
| 12.0(5)T | This command was introduced. |
| 12.2 | Support for named and extend access lists was introduced. |

| Release | Modification |
|---|---|
| 12.3(1) | The following keywords were introduced:<br>• **ftp**<br>• **telnet**<br>• **inactivity-timer** *min*<br>• **absolute-timer** *min* |
| 12.4(6)T | The **service-policy type tag** keywords and *service-policy-name* argument was added. |

## Usage Guidelines

This command creates a named authentication proxy rule, and it allows you to associate that rule with an access control list (ACL), providing control over which hosts use the authentication proxy. The rule is applied to an interface on a router using the **ip auth-proxy** command.

Use the **inactivity-timer** *min* option to override the global the authentication proxy cache timer. This option provides control over timeout values for specific authentication proxy rules. The authentication proxy cache timer monitors the length of time (in minutes) that an authentication cache entry, along with its associated dynamic user access control list, is managed after a period of inactivity. When that period of inactivity (idle time) expires, the authentication entry and the associated dynamic access lists are deleted.

Use the **list** option to associate a set of specific IP addresses or a named ACL with the **ip auth-proxy name** command.

Use the **no** form of this command with a rule name to remove the authentication proxy rules. If no rule is specified, the **no** form of this command removes all the authentication rules on the router, and disables the proxy at all interfaces.

**Note** You must use the **aaa authorization auth-proxy** command together with the **ip auth-proxy name** command. Together these commands set up the authorization policy to be retrieved by the firewall. Refer to the **aaa authorization auth-proxy** command for more information.

## Examples

The following example creates the HQ_users authentication proxy rule. Because an access list is not specified in the rule, all connection-initiating HTTP traffic is subjected to authentication.

```
ip auth-proxy name HQ_users http
```

The following example creates the Mfg_users authentication proxy rule and applies it to hosts specified in ACL 10:

```
access-list 10 192.168.7.0 0.0.0.255
ip auth-proxy name Mfg_users http list 10
```

The following example sets the timeout value for Mfg_users to 30 minutes:

```
access-list 15 any
ip auth-proxy name Mfg_users http inactivity-timer 30 list 15
```

The following example disables the Mfg_users rule:

```
no ip auth-proxy name Mfg_users
```

The following example disables the authentication proxy at all interfaces and removes all the rules from the router configuration:

```
no ip auth-proxy
```

## Related Commands

| Command | Description |
|---|---|
| **aaa authorization** | Sets parameters that restrict network access to a user. |
| **ip auth-proxy (global)** | Sets the authentication proxy idle timeout value (the length of time an authentication cache entry, along with its associated dynamic user ACL, is managed after a period of inactivity). |
| **ip auth-proxy (interface)** | Applies an authentication proxy rule at a firewall interface. |
| **show ip auth-proxy configuration** | Displays the authentication proxy entries or the running authentication proxy configuration. |

# match port-type

To match the access policy on the basis of the port for a class map, use the **match port-type** command in class map configuration mode. To delete the port type, use the **no** form of this command.

**match port-type** {**routed** | **switched**}

**no match port-type** {**routed** | **switched**}

## Syntax Description

| | |
|---|---|
| **routed** | Matches the routed interface. Use this keyword if the class map has to be associated with only a routed interface. |
| **switched** | Matches the switched interface. Use this keyword if the class map has to be associated with only a switched interface. |

## Command Default

Access policy is not matched.

## Command Modes

Class map configuration

## Command History

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

## Usage Guidelines

This command is used because, on the basis of the port on which a user is connecting, the access policies that are applied to it can be different.

## Examples

The following example shows that an access policy has been matched on the basis of the port for a class map:

```
Router (config-cmap)# match port routed
```

## Related Commands

| Command | Description |
|---|---|
| **class-map** | Creates a class map to be used for matching packets to a specified class. |
| **match tag (class-map)** | Specifies the tag to be matched for a tag type of class map. |

# match tag (class-map)

To specify the tag to be matched for a tag type of class map, use the **match tag** command in class map configuration mode. To delete the tag, use the **no** form of this command.

**match tag** {*tag-name*}

**no match tag** {*tag-name*}

## Syntax Description

| | |
|---|---|
| *tag-name* | Name of the tag. |

## Command Default

No match tags are defined.

## Command Modes

Class map configuration

## Command History

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

## Usage Guidelines

The access control server (ACS) sends the tag attribute to the network access device (NAD) using the Cisco attribute-value (AV) pair. (The tag attribute can also be sent to the NAD using the IETF attribute 88.

## Examples

The following example shows that the tag to be matched is named “healthy”:

```
Router (config)# class-map type tag healthy_class
Router(config-cmap)# match tag healthy
Router(config-cmap)# end
```

## Related Commands

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

# policy-map

To create or modify a policy map that can be attached to one or more interfaces to specify a service policy, use the **policy-map** command in global configuration command. To delete a policy map, use the **no** form of this command.

**policy-map** [**type access-control | type control tag**] *policy-map-name*

**no policy-map** [**type access-control | type control tag**] *policy-map-name*

## Syntax Description

| | |
|---|---|
| **type access-control** | (Optional) Determines the exact pattern to look for in the protocol stack of interest. |
| **type control tag** | (Optional) Creates a policy map type tag. |
| *policy-map-name* | Name of the policy map. The name can be a maximum of 40 alphanumeric characters. |

## Defaults

No default behavior or values

## Command Modes

Global configuration

## Command History

| Release | Modification |
|---|---|
| 12.0(5)T | This command was introduced. |
| 12.4(4)T | The **type access-control** keywords were added to support flexible packet matching. |
| 12.4(6)T | The **type control tag** keyword was added. |

## Usage Guidelines

Use the **policy-map** command to specify the name of the policy map to be created, added to, or modified before you can configure policies for classes whose match criteria are defined in a class map. Entering the **policy-map** command enables QoS policy-map configuration mode in which you can configure or modify the class policies for that policy map.

You can configure class policies in a policy map only if the classes have match criteria defined for them. You use the **class-map** and **match** commands to configure the match criteria for a class. Because you can configure a maximum of 64 class maps, no policy map can contain more than 64 class policies.

A single policy map can be attached to multiple interfaces concurrently. When you attempt to attach a policy map to an interface, the attempt is denied if the available bandwidth on the interface cannot accommodate the total bandwidth requested by class policies comprising the policy map. In this case, if the policy map is already attached to other interfaces, it is removed from them.

Whenever you modify class policy in an attached policy map, CBWFQ is notified and the new classes are installed as part of the policy map in the CBWFQ system.

## Examples

The following example creates a policy map called policy1 and configures two class policies included in that policy map. The class policy called class1 specifies policy for traffic that matches access control list (ACL) 136. The second class is the default class to which packets that do not satisfy configured match criteria are directed.

```
! The following commands create class-map class1 and defines its match criteria:
class-map class1
 match access-group 136

! The following commands create the policy map, which is defined to contain policy
! specification for class1 and the default class:
policy-map policy1

class class1
 bandwidth 2000
 queue-limit 40

class class-default
 fair-queue 16
 queue-limit 20
```

The following example creates a policy map called policy9 and configures three class policies to belong to that map. Of these classes, two specify policy for classes with class maps that specify match criteria based on either a numbered ACL or an interface name, and one specifies policy for the default class called **class-default** to which packets that do not satisfy configured match criteria are directed.

```
policy-map policy9

class acl136
  bandwidth 2000
  queue-limit 40

 class ethernet101
  bandwidth 3000
  random-detect exponential-weighting-constant 10

 class class-default
  fair-queue 10
  queue-limit 20Related Commands
```

The following example shows that a policy map type tag has been created:

```
! The following line will be associated with the IP admission name.
Router (config)# policy-map type control tag global_class
! The following line refers to the class map that was defined above.
Router (config-pmap)# class healthy_class
Router (config-pmap-c)# identity policy healthy_policy
Router(config-pmap-c)# exit
The following line refers to the non_healthy class that was defined above.
Router (config-pmap)# class non_healthy_class
Router(config-pmap-c)# identity policy non_healthy_policy
Router (config-pmap-c)# end
```

## Related Commands

| Command | Description |
|---|---|
| **bandwidth (policy-map class)** | Specifies or modifies the bandwidth allocated for a class belonging to a policy map. |
| **class (policy-map)** | Specifies the name of the class whose policy you want to create or change, and the default class (commonly known as the class-default class) before you configure its policy. |

| Command | Description |
|---|---|
| **class class-default** | Specifies the default class whose bandwidth is to be configured or modified. |
| **class-map** | Creates a class map to be used for matching packets to a specified class. |
| **fair-queue (class-default)** | Specifies the number of dynamic queues to be reserved for use by the class-default class as part of the default class policy. |
| **queue-limit** | Specifies or modifies the maximum number of packets the queue can hold for a class policy configured in a policy map. |
| **random-detect (interface)** | Enables WRED or DWRED. |
| **random-detect exponential-weighting-constant** | Configures the WRED and DWRED exponential weight factor for the average queue size calculation. |
| **random-detect precedence** | Configures WRED and DWRED parameters for a particular IP Precedence. |
| **service-policy** | Attaches a policy map to an input interface or VC, or an output interface or VC, to be used as the service policy for that interface or VC. |

# show class-map

To display all class maps and their matching criteria, use the **show class-map** command in privileged EXEC mode.

**show class-map** [**type** {**stack** | **access-control** | **tag**}] [*class-map-name*]

## Syntax Description

| | |
|---|---|
| *class-map-name* | (Optional) Name of the class map. The class map name can be a maximum of 40 alphanumeric characters. |
| **type stack** | (Optional) Displays class maps that are configured to determine the correct protocol stack in which to examine via flexible packet matching (FPM). |
| **type access-control** | (Optional) Displays class maps that are configured to determine the exact pattern to look for in the protocol stack of interest. |
| **type tag** | (Optional) Displays class maps that are configured to determine the class-map configuration. |

## Command Modes

Privileged EXEC

## Command History

| Release | Modification |
|---|---|
| 12.0(5)T | This command was introduced. |
| 12.2(13)T | This command was modified to display the Frame Relay data-link connection identified (DLCI) number as a criterion for matching traffic inside a class map.<br>In addition, this command was modified to display Layer 3 packet length as a criterion for matching traffic inside a class map. |
| 12.4(4)T | The **type**, **stack**, and **access-control** keywords were added to support flexible packet matching (FPM). |
| 12.2(18)SXE | This command was integrated into Cisco IOS Release 12.2(18)SXE. |
| 12.2(28)SB | This command was integrated into Cisco IOS Release 12.2(28)SB. |
| 12.4(6)T | The **tag** keyword was added. |

## Usage Guidelines

You can use the **show class-map** command to display all class maps and their matching criteria. If you enter the optional *class-map-name* argument, the specified class map and its matching criteria will be displayed.

## Examples

In the following example, three class maps are defined. Packets that match access list 103 belong to class c3, IP packets belong to class c2, and packets that come through input Ethernet interface 1/0 belong to class c1. The output from the **show class-map** command shows the three defined class maps.

```
Router# show class-map

 Class Map c3
 Match access-group 103
```

```
 Class Map c2
 Match protocol ip

 Class Map c1
 Match input-interface Ethernet1/0
```

In the following example, a class map called “c1” has been defined, and the Frame Relay DLCI number of 500 has been specified as a match criterion:

```
Router# show class-map

class map match-all c1
   match fr-dlci 500
```

The following example shows that the **type tag** keyword has been used to determine the class-map configuration.

```
Router# show class-map type tag

class map type tag match-all temp (id 1)
   match tag healthy
```

Table 1 describes the significant fields shown in the display.

***Table 1 show class-map Field Descriptions[1]***

| Field | Description |
|---|---|
| Class-map | Class of traffic being displayed. Output is displayed for each configured class map in the policy. The choice for implementing class matches (for example, match-all or match-any) can also appear next to the traffic class. |
| Match | Match criteria specified for the class map. Choices include criteria such as the Frame Relay DLCI number, Layer 3 packet length, IP precedence, IP differentiated services code point (DSCP) value, Multiprotocol Label Switching (MPLS) experimental value, access groups, and quality of service (QoS) groups. |

1. A number in parentheses may appear next to the class-map name, and match criteria information. The number is for Cisco internal use only and can be disregarded.

## Related Commands

| Command | Description |
|---|---|
| **class-map** | Creates a class map to be used for matching packets to a specified class. |
| **match fr-dlci** | Specifies the Frame Relay DLCI number as a match criterion in a class map. |
| **match packet length (class-map)** | Specifies and uses the length of the Layer 3 packet in the IP header as a match criterion in a class map. |
| **show policy-map** | Displays the configuration of all classes for a specified service policy map or all classes for all existing policy maps. |
| **show policy-map interface** | Displays the packet statistics of all classes that are configured for all service policies either on the specified interface or subinterface or on a specific PVC on the interface. |

# show epm session ip

To display whether tag policies have been applied, use the **show epm session ip** command in privileged EXEC mode.

**show epm session ip** {*ip-address* | **summary**}

## Syntax Description

| | |
|---|---|
| *ip-address* | Information is displayed for a specific IP address. |
| **summary** | Information is displayed for all sessions. |

## Command Modes

Privileged EXEC

## Command History

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

## Examples

The following example shows information for all sessions:

```
Router# show epm session ip summary

Total sessions seen so far : 1
Total active sessions      : 1
Session IP Address : 10.9.0.1
```

The following output shows information specifically for IP address 10.9.0.1

```
Router# show epm session ip 10.9.0.1

Admission feature       : Eapoudp
Tag Received            : healthy
Policy map used         : temp
Class map matched       : temp
```

Table 2 describes significant fields shown in the displays.

***Table 2 show epm session ip Field Descriptions***

| Field | Description |
|---|---|
| Total sessions seen so far | Total number of hosts connected to NAD till now. |
| Total active sessions | Total number of active sessions. |
| Session IP Address | Active session information. |
| Admission feature | Admission feature authentication proxy or Extensible Authentication Protocol over UDP (EOU) acting on the host. |
| Tag Received | Tag attribute received from the access control server (ACS). |

*Table 2 show epm session ip Field Descriptions (continued)*

| | |
|---|---|
| Policy map used | Policy map associated with the IP admission rule. |
| Class map matched | Class map in the policy map that matched. |

# show policy-map

To display the configuration of all classes for a specified service policy map or all classes for all existing policy maps, use the **show policy-map** command in privileged EXEC mode.

**show policy-map** [*policy-map*] [**type control tag**]

## Syntax Description

| | |
|---|---|
| *policy-map* | (Optional) Name of the service policy map whose complete configuration is to be displayed. |
| **type control tag** | (Optional) Displays information about the policy map type tag. |

## Defaults

All existing policy map configurations are displayed.

## Command Modes

Privileged EXEC

## 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. |
| 12.1(1)E | This command was integrated into Cisco IOS Release 12.1(1)E. |
| 12.2(13)T | The output of this command was modified for the Percentage-Based Policing and Shaping feature and includes the bandwidth percentage used when calculating traffic policing and shaping. |
| 12.0(28)S | The output of this command was modified for the QoS: Percentage-Based Policing feature to display the committed (conform) burst (bc) and excess (peak) burst (be) sizes in milliseconds (ms). |
| 12.2(18)SXE | This command was integrated into Cisco IOS Release 12.2(18)SXE. |
| 12.4(6)T | The **type control tag** keyword was added. |
| 12.2(28)SB | This command was integrated into Cisco IOS Release 12.2(28)SB and its output was modified to display class-based policies when using hierarchical queueing framework (HQF) on an interface. |

## Usage Guidelines

The **show policy-map** command displays the configuration of a service policy map created using the **policy-map** command. You can use the **show policy-map** command to display all class configurations comprising any existing service policy map, whether or not that service policy map has been attached to an interface.

## Examples

The following is sample output from the **show policy-map** command. This sample output displays the contents of a policy map called "policy1." In policy 1, traffic policing on the basis of a committed information rate (CIR) of 20 percent has been configured, and the bc and be have been specified in milliseconds. As part of the traffic policing configuration, optional conform, exceed, and violate actions have been specified.

```
Router# show policy-map policy1
  Policy Map policy1
    Class class1
     police cir percent 20 bc 300 ms pir percent 40 be 400 ms
       conform-action transmit
       exceed-action drop
       violate-action drop
```

The following example shows that the type control tag is to be displayed:

```
Router# show policy-map type control tag

  Policy Map type tag temp
    Class temp
```

Table 3 describes the significant fields shown in the display.

***Table 3 show policy-map Field Descriptions***

| Field | Description |
|---|---|
| Policy Map | Name of policy map displayed. |
| Class | Name of class configured in policy map displayed. |
| police | Indicates that traffic policing on the basis of specified percentage of bandwidth has been enabled. The committed burst (bc) and excess burst (be) sizes have been specified in milliseconds (ms), and optional conform, exceed, and violate actions have been specified. |

## Related Commands

| Command | Description |
|---|---|
| **policy-map** | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |
| **show policy-map class** | Displays the configuration for the specified class of the specified policy map. |
| **show policy-map interface** | Displays the packet statistics of all classes that are configured for all service policies either on the specified interface or subinterface or on a specific PVC on the interface. |

# Feature Information for Tag and Template

Table 4 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

**Note** Table 4 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

*Table 4 Feature Information for Tag and Template*

| Feature Name | Releases | Feature Information |
|---|---|---|
| Tag and Template | 12.4(6)T | The Tag and Template feature allows network administrators to define enforcement policies on a local device and have a RADIUS server specify the policy selector to be enforced. This feature can be applied to a Network Admission Control (NAC) architecture.<br><br>The following commands were introduced or modified by this feature: **class-map**, **class type**, **debug tag-template event**, **identity policy (policy-map)**, **ip admission name**, **ip auth-proxy name**, **match port-type**, **match tag (class-map)**, **show class-map**, and **show policy-map type**. |