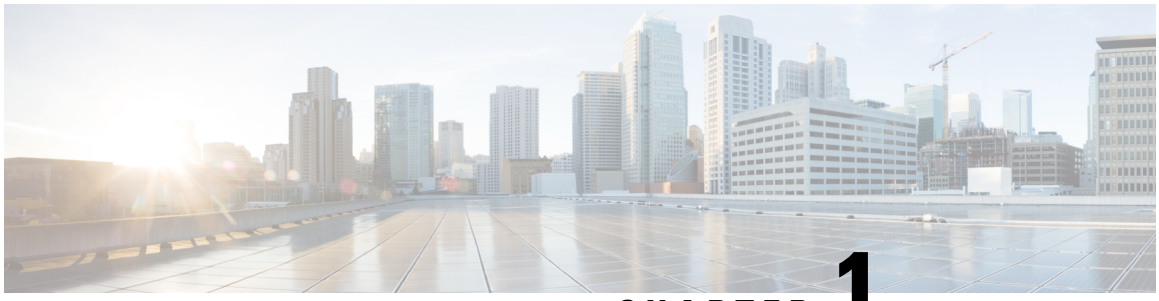
Identity-Based Networking Services Configuration Guide, Cisco IOS XE Release 3E

First Published: 2013-01-29

Last Modified: 2013-01-29

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Identity-Based Networking Services Overview

Identity-Based Networking Services provides a policy and identity-based framework in which edge devices can deliver flexible and scalable services to subscribers. This module provides information about what Identity-Based Networking Services is and its features and benefits.

Finding Feature Information

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

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

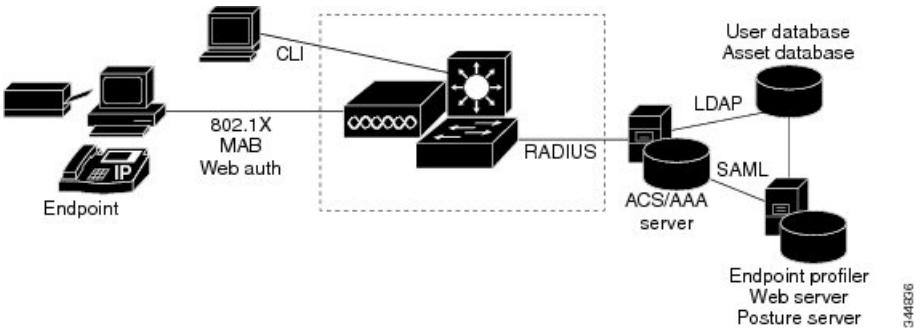
Information About Identity-Based Networking Services

Understanding Identity-Based Networking Services

Identity-Based Networking Services provides an identity-based approach to access management and subscriber management. It offers a consistent way to configure features across technologies, a command interface that allows easy deployment and customization of features, and a robust policy control engine with the ability to apply policies defined locally or received from an external server to enforce policy in the network.

The figure below illustrates a typical deployment of Identity-Based Networking Services in a physically distributed enterprise with a campus, branch offices, and remote workers.

Figure 1: Sample Deployment



Features in Identity-Based Networking Services

Identity-Based Networking Services includes the following features:

- Cisco common classification policy language (C3PL)-based identity configuration
- Concurrent authentication methods on a single session, including IEEE 802.1x (dot1x), MAC authentication bypass (MAB), and web authentication
- Downloadable identity service templates
- Extended RADIUS change of authorization (CoA) support for querying, reauthenticating, and terminating a session, port shutdown and port bounce, and activating and deactivating an identity service template.
- Local authentication using Lightweight Directory Access Protocol (LDAP)
- Locally defined identity control policies
- Locally defined identity service templates
- Per-user inactivity handling across methods
- Web authentication support of common session ID
- Web authentication support of IPv6

Benefits of Identity-Based Networking Services

Identity-based solutions are essential for delivering access control for disparate groups such as employees, contractors, and partners while maintaining low operating expenses. Identity-Based Networking Services provides a consistent approach to operational management through a policy and identity-based infrastructure leading to faster deployment of new features and easier management of switches.

Identity-Based Networking Services provides the following benefits:

- An identity-based framework for session management.
- A robust policy control engine to apply policies defined locally or received from an external AAA server.
- Faster deployment and customization of features across access technologies.

- A simpler and consistent way to configure features across access methods, platforms, and application domains.

Web Authentication Support for Common Session ID

Identity-Based Networking Services allows a single session identifier to be used for web authentication sessions in addition to all 802.1X and MAB authenticated sessions for a client. This session ID is used for all reporting purposes such as show commands, MIBs, and RADIUS messages and allows users to distinguish messages for one session from messages for other sessions. This common session ID is used consistently across all authentication methods and features applied to a session.

Web Authentication Support of IPv6

Identity-Based Networking Services introduces IPv6 support for web authentication. IPv6 is supported for web authentication only when Identity-Based Networking Services is explicitly configured. This means that you must permanently convert your configuration to the Cisco common classification policy language (C3PL) display mode by specifically configuring a Identity-Based Networking Services command such as the **policy-map type control subscriber** command.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Identity-Based Networking Services commands	Cisco IOS Identity-Based Networking Services Command Reference
Address Resolution Protocol (ARP) commands	Cisco IOS IP Addressing Services Command Reference
ARP configuration tasks	IP Addressing - ARP Configuration Guide
Authentication, authorization, and accounting (AAA) configuration tasks	Authentication Authorization and Accounting Configuration Guide
AAA commands	Cisco IOS Security Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 5176	Dynamic Authorization Extensions to RADIUS

Technical Assistance

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

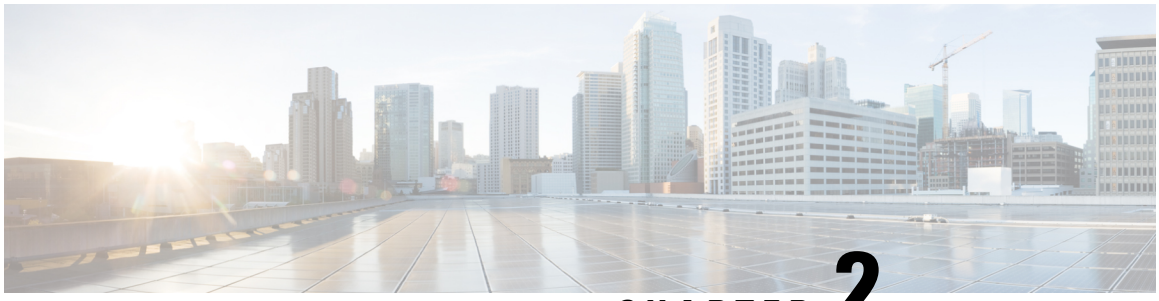
Feature Information for Identity-Based Networking Services Overview

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

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

Table 1: Feature Information for Identity-Based Networking Services Overview

Feature Name	Releases	Feature Information
Web Authentication Support of Common Session ID	Cisco IOS XE Release 3.2SE Cisco IOS XE Release 3.6E	Allows a single session identifier to be used for all web authentication sessions in addition to 802.1X and MAB authenticated sessions. In Cisco IOS XE 3.2SE, this feature is supported on the following platforms: • Cisco Catalyst 3650 Series Switches • Cisco Catalyst 3850 Series Switches • Cisco Catalyst 4500E Supervisor Engine 6-E • Cisco Catalyst 4500E Supervisor Engine 6L-E • Cisco Catalyst 4500E Supervisor Engine 7-E • Cisco Catalyst 4500E Supervisor Engine 7L-E • Cisco 5700 Wireless LAN Controllers In Cisco IOS XE Release 3.6E, this feature is supported on Cisco Catalyst 3850 Series Switches.



CHAPTER 2

Change of Authorization Support

Identity-Based Networking Services supports RADIUS change of authorization (CoA) commands for session query, reauthentication, and termination, port bounce and port shutdown, and service template activation and deactivation. This module provides information about the supported CoA commands for Identity-Based Networking Services.

Finding Feature Information

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

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

Information About CoA Support

RADIUS Change-of-Authorization Support

Cisco IOS software supports the RADIUS CoA extensions defined in RFC 5176 that are typically used in a push model to allow the dynamic reconfiguring of sessions from external AAA or policy servers. Per-session CoA requests are supported for session identification, session termination, host reauthentication, port shutdown, and port bounce. This model comprises one request (CoA-Request) and two possible response codes:

- CoA acknowledgement (ACK) [CoA-ACK]
- CoA nonacknowledgement (NAK) [CoA-NAK]

The request is initiated from a CoA client (typically a AAA or policy server) and directed to the device that acts as a listener.

The table below shows the RADIUS CoA commands and vendor-specific attributes (VSAs) supported by Identity-Based Networking Services. All CoA commands must include the session identifier between the device and the CoA client.

Table 2: RADIUS CoA Commands Supported by Identity-Based Networking Services

CoA Command	Cisco VSA
Activate service	Cisco:Avpair="subscriber:command=activate-service" Cisco:Avpair="subscriber:service-name=<*service-name*>" Cisco:Avpair="subscriber:precedence=<*precedence-number*>" Cisco:Avpair="subscriber:activation-mode=replace-all"
Deactivate service	Cisco:Avpair="subscriber:command=deactivate-service" Cisco:Avpair="subscriber:service-name=<*service-name*>"
Bounce host port	Cisco:Avpair="subscriber:command=bounce-host-port"
Disable host port	Cisco:Avpair="subscriber:command=disable-host-port"
Session query	Cisco:Avpair="subscriber:command=session-query"
Session reauthenticate	Cisco:Avpair="subscriber:command=reauthenticate" Cisco:Avpair="subscriber:reauthenticate-type=last" or Cisco:Avpair="subscriber:reauthenticate-type=rerun"
Session terminate	This is a standard disconnect request and does not require a VSA.
Interface template	Cisco:AVpair="interface-template-name=<*interfacetemplate*>"

Session Identification

For disconnect and CoA requests targeted at a particular session, the device locates the session based on one or more of the following attributes:

- Acct-Session-Id (IETF attribute #44)
- Audit-Session-Id (Cisco VSA)
- Calling-Station-Id (IETF attribute #31, which contains the host MAC address)
- IPv6 Attributes, which can be one of the following:
  - Framed-IPv6-Prefix (IETF attribute #97) and Framed-Interface-Id (IETF attribute #96), which together create a full IPv6 address per RFC 3162
  - Framed-IPv6-Address
- Plain IP Address (IETF attribute #8)

If more than one session identification attribute is included in the message, all of the attributes must match the session or the device returns a Disconnect-NAK or CoA-NAK with the error code "Invalid Attribute Value."

For CoA requests targeted at a particular enforcement policy, the device returns a CoA-NAK with the error code "Invalid Attribute Value" if any of the above session identification attributes are included in the message.

CoA Activate Service Command

The CoA activate service command can be used to activate a service template on a session. The AAA server sends the request in a standard CoA-Request message using the following VSAs:

Cisco:Avpair="subscriber:command=activate-service"

Cisco:Avpair="subscriber:service-name=<*service-name*>"

Cisco:Avpair="subscriber:precedence=<*precedence-number*>"

Cisco:Avpair="subscriber:activation-mode=replace-all"

Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes described in the "Session Identification, on page 6" section. If the device cannot locate a session, it returns a CoA-NAK message with the "Session Context Not Found" error-code attribute. If the device locates a session, it initiates an activate template operation for the hosting port and a CoA-ACK is returned. If activating the template fails, a CoA-NAK message is returned with the Error-Code attribute set to the appropriate message.

If the device fails before returning a CoA-ACK to the client, the process is repeated on the new active device when the request is re-sent from the client. If the device fails after returning a CoA-ACK message to the client but before the operation is complete, the operation is restarted on the new active device.

CoA Deactivate Service Command

The CoA deactivate service command can be used to deactivate a service template on a session. The AAA server sends the request in a standard CoA-Request message using the following VSAs:

Cisco:Avpair="subscriber:command=deactivate-service"

Cisco:Avpair="subscriber:service-name=<*service-name*>"

Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes described in the "Session Identification, on page 6" section. If the device cannot locate a session, it returns a CoA-NAK message with the "Session Context Not Found" error-code attribute. If the device locates a session, it initiates a deactivate template operation for the hosting port and a CoA-ACK is returned. If deactivating the template fails, a CoA-NAK message is returned with the Error-Code attribute set to the appropriate message.

If the device fails before returning a CoA-ACK to the client, the process is repeated on the new active device when the request is re-sent from the client. If the device fails after returning a CoA-ACK message to the client but before the operation is complete, the operation is restarted on the new active device.

CoA Bounce Host Port Command

The CoA bounce host port command terminates a session and bounces the port (initiates a link down event followed by a link up event). The AAA server sends the request in a standard CoA-Request message with the following VSA:

Cisco:Avpair="subscriber:command=bounce-host-port"

Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes described in the "Session Identification, on page 6" section. If the session cannot be located, the device returns a CoA-NAK message with the "Session Context Not Found" error-code attribute. If the session is located, the device disables the hosting port for a period of ten seconds, reenables it (port bounce), and returns a CoA-ACK.

If the device fails before returning a CoA-ACK to the client, the process is repeated on the new active device when the request is re-sent from the client. If the device fails after returning a CoA-ACK message to the client but before the operation is complete, the operation is restarted on the new active device.

The CoA bounce port command is useful as a last resort when an endpoint needs to acquire a new IP address after a change in authorization and this is the only way to indicate to the endpoint to restart the DHCP process. This can occur when there is a VLAN change and the endpoint is a device, such as a printer, that does not have a mechanism to detect a change on this authentication port. This command can cause a link flap on an authentication port, which triggers DHCP renegotiation from one or more hosts connected to this port.

CoA Disable Host Port Command

The CoA disable host port command administratively shuts down the authentication port that is hosting a session, which terminates the session. The AAA server sends the request in a standard CoA-Request message with the following VSA:

Cisco:Avpair="subscriber:command=disable-host-port"

Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes described in the "Session Identification, on page 6" section. If the device cannot locate the session, it returns a CoA-NAK message with the "Session Context Not Found" error-code attribute. If the device locates the session, it disables the hosting port and returns a CoA-ACK message.

If the device fails before returning a CoA-ACK to the client, the process is repeated on the new active device when the request is re-sent from the client. If the device fails after returning a CoA-ACK message to the client but before the operation is complete, the operation is restarted on the new active device.

CoA Session Query Command

The CoA session query command requests service information about a subscriber session. The AAA server sends the request in a standard CoA-Request message containing the following VSA:

Cisco:Avpair="subscriber:command=session-query"

Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes described in the "Session Identification, on page 6" section. If the device cannot locate a session, it returns a CoA-NAK message with the "Session Context Not Found" error-code attribute. If the device locates a session, it performs a session query operation on the session and returns a CoA-ACK message.

If the device fails before returning a CoA-ACK to the client, the process is repeated on the new active device when the request is re-sent from the client. If the device fails after returning a CoA-ACK message to the client but before the operation is complete, the operation is restarted on the new active device.

CoA Session Reauthenticate Command

To initiate session authentication, the AAA server sends a standard CoA-Request message containing the following VSAs:

Cisco:Avpair="subscriber:command=reauthenticate"

Cisco:Avpair="subscriber:reauthenticate-type=<last | rerun>"

"reauthenticate-type" defines whether the CoA reauthentication request uses the authentication method that last succeeded on the session or whether the authentication process is completely rerun.

The following rules apply:

- "subscriber:command=reauthenticate" must be present to trigger a reauthentication.
- If "subscriber:reauthenticate-type" is not specified, the default behavior is to rerun the last successful authentication method for the session. If the method reauthenticates successfully, all old authorization data is replaced with the new reauthenticated authorization data.
- "subscriber:reauthenticate-type" is valid only when included with "subscriber:command=reauthenticate." If it is included in another CoA command, the VSA will be silently ignored.

If the device fails before returning a CoA-ACK to the client, the process is repeated on the new active device when the request is resent from the client. If the device fails after returning a CoA-ACK message to the client but before the operation is complete, the operation is restarted on the new active device.

CoA Session Terminate Command

A CoA Disconnect-Request command terminates a session without disabling the host port. This command causes reinitialization of the authenticator state machine for the specified host, but does not restrict the host's access to the network. If the session cannot be located, the device returns a Disconnect-NAK message with the "Session Context Not Found" error-code attribute. If the session is located, the device terminates the session. After the session has been completely removed, the device returns a Disconnect-ACK.

If the device fails before returning a CoA-ACK to the client, the process is repeated on the new active device when the request is re-sent from the client.

To restrict a host's access to the network, use a CoA Request with the Cisco:Avpair="subscriber:command=disable-host-port" VSA. This command is useful when a host is known to cause problems on the network and network access needs to be immediately blocked for the host. When you want to restore network access on the port, reenable it using a non-RADIUS mechanism.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Identity-Based Networking Services commands	Cisco IOS Identity-Based Networking Services Command Reference
Address Resolution Protocol (ARP) commands	Cisco IOS IP Addressing Services Command Reference
ARP configuration tasks	IP Addressing - ARP Configuration Guide
Authentication, authorization, and accounting (AAA) configuration tasks	Authentication Authorization and Accounting Configuration Guide
AAA commands	Cisco IOS Security Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 5176	Dynamic Authorization Extensions to RADIUS

Technical Assistance

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

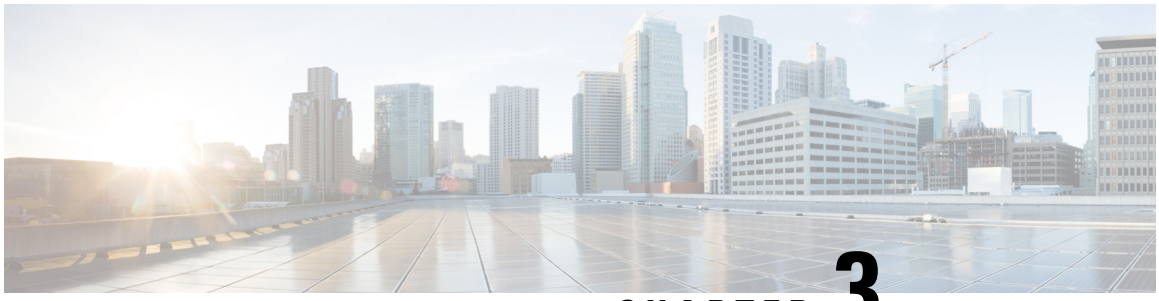
Feature Information for CoA Support

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

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

Table 3: Feature Information for CoA Support

Feature Name	Releases	Feature Information
Change of Authorization	Cisco IOS XE Release 3.2SE	Supports CoA requests for initiating the following: • Activating and deactivating service templates on sessions • Port bounce • Port shutdown • Querying a session • Reauthenticating a session • Terminating a session These VSAs are sent in a standard CoA-Request message from a AAA server. In Cisco IOS XE Release 3.2SE, this feature is supported on the following platforms: • Cisco 5700 Series Wireless LAN Controllers • Cisco Catalyst 3850 Series Switches In Cisco IOS XE Release 3.3SE, this feature is supported on the following platforms: • Cisco Catalyst 3650 Series Switches In Cisco IOS XE 3.5E, this feature is supported on the following platforms: • Cisco Catalyst 4500E Supervisor Engine 6-E • Cisco Catalyst 4500E Supervisor Engine 6L-E • Cisco Catalyst 4500E Supervisor Engine 7-E • Cisco Catalyst 4500E Supervisor Engine 7L-E



CHAPTER 3

Configuring Identity Control Policies

Identity control policies define the actions that Identity-Based Networking Services takes in response to specified conditions and subscriber events. A variety of system actions, conditions, and events can be combined using a consistent policy language. This module provides information about how to configure identity control policies for Identity-Based Networking Services.

Finding Feature Information

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

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

Information About Identity Control Policies

Identity-Based Networking Services Configuration

To convert all relevant authentication commands to their Class-Based Policy Language(CPL) control policy equivalents, use the **authentication convert-to new-style** command. This command permanently converts the legacy configuration on the switch to identity-based networking services.

Note This configuration is irreversible. It disables the conversion command – **authentication display [legacy | new-style]**.

Use the **authentication display config-mode** command in EXEC mode to display the current configuration mode; *legacy* if it is legacy mode and **new-style** if it is Identity-Based Networking Services configuration mode.

```
(Device)# authentication display config-mode
Current configuration mode is legacy

Device)# authentication display config-mode
Current configuration mode is new-style
```

Concurrent Authentication Methods

Identity-Based Networking Services allows the concurrent operation of IEEE 802.1x (dot1x), MAC authentication bypass (MAB), and web authentication methods, making it possible to invoke multiple authentication methods in parallel on a single subscriber session. This allows the client-supported method to complete at the earliest opportunity without the delays associated with serialization.

Typically, the access control method that is used to authorize a host is left up to the endpoint. For example, a printer without an 802.1x supplicant would be authorized through MAB only, an employee desktop through 802.1x only, and a guest through web authentication only. The default priority order is 802.1x, followed by MAB, then web authentication. When method priorities are the same, the first method that successfully authenticates the session prevails.

An example in which more than one method may succeed during the lifetime of a session is when MAB is used to provide interim access pending success of 802.1x. A host could be also be given interim access to a web server to allow credentials to be updated so that 802.1x can succeed after an authentication failure.

Configuration Display Mode

Identity-Based Networking Services introduces new Cisco IOS commands that replace many of the previously supported authentication and policy commands. These commands are available only after enabling the Cisco common classification policy language (C3PL) display mode that supports Identity-Based Networking Services. Identity-Based Networking Services features such as concurrent authentication and web authentication with IPv6 are not supported in legacy mode.

The device defaults to the legacy configuration mode until you do one of the following:

- Enter the **authentication display new-style** command—This command switches to C3PL display mode, temporarily converting your legacy configuration to a Identity-Based Networking Services configuration so you can see how it looks before you make the conversion permanent. You can switch back to legacy mode by using the **authentication display legacy** command. See the "Enabling the Display Mode for Identity-Based Networking Services, on page 16" section.
- Enter a Identity-Based Networking Services configuration command—After you enter the first explicit Identity-Based Networking Services command, the configuration converts to C3PL display mode permanently and legacy commands are suppressed. The **authentication display** command is disabled and you can no longer revert to the legacy configuration mode.

Control Policies for Identity-Based Networking Services

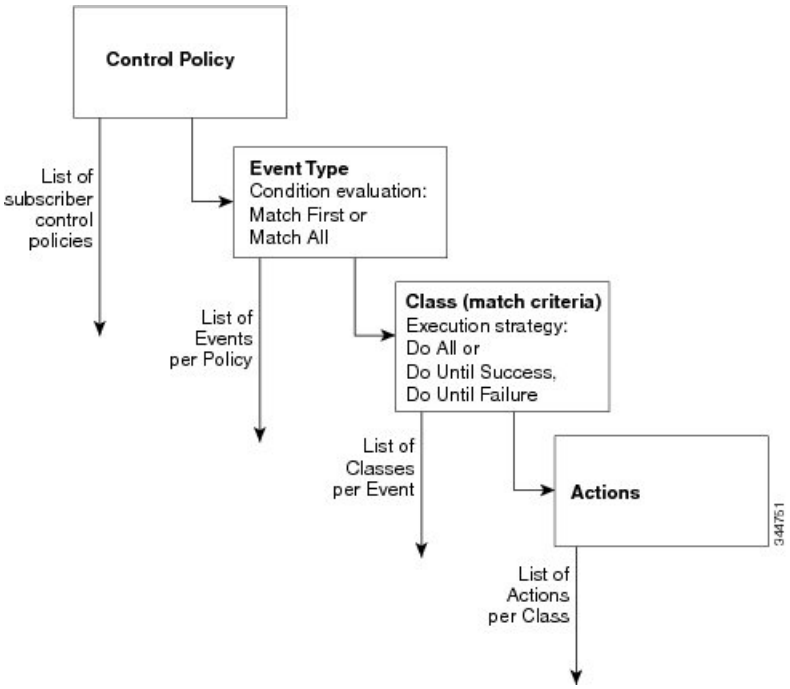
A control policy defines the handling of different subscriber life-cycle events. For various events, such as session start or session failure, you can specify actions in the control policy. These actions can be executed conditionally for different subscribers based on various match criteria. Control policies are activated on

interfaces and typically control the authentication of subscriber identity and the activation of services on sessions. For example, you can configure a control policy to authenticate specific subscribers and then provide them with access to specific services.

A control policy consists of one or more control policy rules and a decision strategy that governs how the policy rules are evaluated. A control policy rule consists of a control class (a flexible condition clause), an event for which the condition is evaluated, and one or more actions. Actions are general system functions, such as “authenticate” or “activate.” You define the specific actions that an event will trigger and some events have default actions.

The figure below illustrates how each control policy contains a list of events that are considered applicable to the subscriber life cycle. Within each event type is a list of control classes with different match criteria for subscriber identity, and under each class is a list of actions to be executed.

Figure 2: Control Policy Structure



Control Policy Configuration Overview

Control policies express system functionality in terms of an event, a condition, and an action. There are three steps in defining a control policy:

1. Create one or more control classes—A control class specifies the conditions that must be met for a control policy to be activated. A control class can contain multiple conditions, each of which will evaluate as either true or false. Match directives specify whether all, any, or none of the individual conditions must evaluate true for the class to evaluate true. Or, you can specify the default control class which does not contain any conditions and always evaluates true.
2. Create a control policy—A control policy contains one or more control policy rules. A control policy rule consists of a control class, an event that causes the class to be evaluated, and one or more actions. Actions are numbered and executed sequentially.

3. Apply the control policy—A control policy is activated by applying it to an interface.

Parameter Maps for Identity-Based Networking Services

A parameter map allows you to specify parameters that control the behavior of actions specified under a control policy. For Identity-Based Networking Services, an authentication parameter map defines parameters used for the action specified with the **authenticate using webauth** command. You can configure the following types of parameter maps:

- Authentication bypass (This is also called nonresponsive host [NRH] authentication.)
- Consent
- Web authentication
- Web authentication with consent

Parameter maps are optional. If you do not configure a named parameter map, the software uses the default parameters that are specified in the global parameter map.

Per User Inactivity Handling Across Methods

A common inactivity aging feature extends support for RADIUS attributes 28 (Idle-Timeout) and attribute 29 (Termination-Action) to web authenticated sessions, providing consistent inactivity handling across all authentication methods, including 802.1x, MAC authentication bypass (MAB), and web authentication. The AAA server sends these attributes as part of the user authorization. After a session has been idle for the amount of time specified in attribute 28, or has reached the timeout configured with attribute 29, the session is terminated.

You can also apply the inactivity timeout and absolute timeout to sessions through a locally defined service template. When enabling the inactivity timeout, you can also enable address resolution protocol (ARP) probes that are sent before the session is terminated. For configuration information, see the "Configuring Identity Service Templates, on page 47" module.

How to Configure Identity Control Policies

Enabling the Display Mode for Identity-Based Networking Services

Identity-Based Networking Services features are configured in the Cisco common classification policy language (C3PL) display mode. The legacy authentication manager mode is enabled by default. You can use the following procedure to switch to C3PL display mode and temporarily convert any legacy configuration commands to their C3PL equivalents. This allows you to preview your legacy configuration as a Identity-Based Networking Services configuration before making the conversion permanent. After you enter an explicit Identity-Based Networking Services command, the conversion becomes permanent and you can no longer revert to legacy mode.

SUMMARY STEPS

1. **enable**

2. **authentication display** {**legacy** | **new-style**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**authentication display** {**legacy**	**new-style**} **Example:** `Device# authentication display new-style`	Sets the display mode for authentication and policy configuration. • The default display mode is legacy. • You can use this command to switch between legacy and C3PL display mode until you execute the first explicit Identity-Based Networking Services command. After you enter the first explicit Identity-Based Networking Services command, for example when configuring a control class or control policy, the system displays a prompt to confirm whether you want to continue because this command will be disabled and you cannot revert to legacy mode. **Note** If you save the configuration while the new-style mode is enabled, and then perform a reload, the display mode is permanently set to new-style. The **authentication display** command is disabled and you cannot revert to legacy mode. For the stack devices and standalone devices to revert to legacy mode, save the new-style configuration in a flash, **write erase** the device and then perform a **reload**.

Configuring a Control Class

A control class defines the conditions under which the actions of a control policy are executed. You define whether all, any, or none of the conditions must evaluate true to execute the actions of the control policy. Control classes are evaluated based on the event specified in the control policy.

Note This procedure shows all of the match conditions that you can configure in a control class. You must specify at least one condition in a control class to make it valid. All other conditions, and their corresponding steps, are optional (steps 4 through 18 below).

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **class-map type control subscriber** {**match-all** | **match-any** | **match-none**} *control-class-name*
4. {**match** | **no-match**} **activated-service-template** *template-name*
5. {**match** | **no-match**} **authorization-status** {**authorized** | **unauthorized**}
6. {**match** | **no-match**} **authorizing-method-priority** {**eq** | **gt** | **lt**} *priority-value*
7. {**match** | **no-match**} **client-type** {**data** | **switch** | **video** | **voice**}
8. {**match** | **no-match**} **current-method-priority** {**eq** | **gt** | **lt**} *priority-value*
9. {**match** | **no-match**} **ip-address** *ip-address*
10. {**match** | **no-match**} **ipv6-address** *ipv6-address*
11. {**match** | **no-match**} **mac-address** *mac-address*
12. {**match** | **no-match**} **method** {**dot1x** | **mab** | **webauth**}
13. {**match** | **no-match**} **port-type** {**l2-port** | **l3-port** | **dot11-port**}
14. {**match** | **no-match**} **result-type** [**method** {**dot1x** | **mab** | **webauth**}] *result-type*
15. {**match** | **no-match**} **service-template** *template-name*
16. {**match** | **no-match**} **tag** *tag-name*
17. {**match** | **no-match**} **timer** *timer-name*
18. {**match** | **no-match**} **username** *username*
19. **end**
20. **show class-map type control subscriber** {**all** | **name** *control-class-name*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map type control subscriber** {**match-all** | **match-any** | **match-none**} *control-class-name* **Example:** `Device(config)# class-map type control subscriber match-all DOT1X_NO_AGENT`	Creates a control class and enters control class-map filter mode. • **match-all**—All of the conditions in the control class must evaluate true. • **match-any**—At least one of the conditions in the control class must evaluate true. • **match-none**—All of the conditions in the control class must evaluate false.
Step 4	{**match** | **no-match**} **activated-service-template** *template-name* **Example:** `Device(config-filter-control-classmap)# match activated-service-template SVC_1`	(Optional) Creates a condition that evaluates true based on the service template activated on a session.

	Command or Action	Purpose
Step 5	{**match** | **no-match**} **authorization-status** {**authorized** | **unauthorized**} **Example:** `Device(config-filter-control-classmap)# match authorization-status authorized`	(Optional) Creates a condition that evaluates true based on a session's authorization status.
Step 6	{**match** | **no-match**} **authorizing-method-priority** {**eq** | **gt** | **lt**} *priority-value* **Example:** `Device(config-filter-control-classmap)# match authorizing-method-priority eq 10`	(Optional) Creates a condition that evaluates true based on the priority of the authorization method. • **eq**—Current priority is equal to *priority-value*. • **gt**—Current priority is greater than *priority-value*. • **lt**—Current priority is less than *priority-value*. • *priority-value*—Priority value to match. Range: 1 to 254, where 1 is the highest priority and 254 is the lowest.
Step 7	{**match** | **no-match**} **client-type** {**data** | **switch** | **video** | **voice**} **Example:** `Device(config-filter-control-classmap)# match client-type data`	(Optional) Creates a condition that evaluates true based on an event's device type.
Step 8	{**match** | **no-match**} **current-method-priority** {**eq** | **gt** | **lt**} *priority-value* **Example:** `Device(config-filter-control-classmap)# match current-method-priority eq 10`	(Optional) Creates a condition that evaluates true based on the priority of the current authentication method.
Step 9	{**match** | **no-match**} **ip-address** *ip-address* **Example:** `Device(config-filter-control-classmap)# match ip-address 10.10.10.1`	(Optional) Creates a condition that evaluates true based on an event's source IPv4 address.
Step 10	{**match** | **no-match**} **ipv6-address** *ipv6-address* **Example:** `Device(config-filter-control-classmap)# match ipv6-address FE80::1`	(Optional) Creates a condition that evaluates true based on an event's source IPv6 address.
Step 11	{**match** | **no-match**} **mac-address** *mac-address* **Example:** `Device(config-filter-control-classmap)# match mac-address aabb.cc00.6500`	(Optional) Creates a condition that evaluates true based on an event's MAC address.
Step 12	{**match** | **no-match**} **method** {**dot1x** | **mab** | **webauth**} **Example:**	(Optional) Creates a condition that evaluates true based on an event's authentication method.

	Command or Action	Purpose
	`Device(config-filter-control-classmap)# match method dot1x`	
Step 13	{**match** | **no-match**} **port-type** {**l2-port** | **l3-port** | **dot11-port**} **Example:** `Device(config-filter-control-classmap)# match port-type l2-port`	(Optional) Creates a condition that evaluates true based on an event's interface type.
Step 14	{**match** | **no-match**} **result-type** [**method** {**dot1x** | **mab** | **webauth**}] *result-type* **Example:** `Device(config-filter-control-classmap)# match result-type agent-not-found`	(Optional) Creates a condition that evaluates true based on the specified authentication result. • To display the available result types, use the question mark (**?**) online help function.
Step 15	{**match** | **no-match**} **service-template** *template-name* **Example:** `Device(config-filter-control-classmap)# match service-template svc_1`	(Optional) Creates a condition that evaluates true based on an event's service template.
Step 16	{**match** | **no-match**} **tag** *tag-name* **Example:** `Device(config-filter-control-classmap)# match tag tag_1`	(Optional) Creates a condition that evaluates true based on the tag associated with an event.
Step 17	{**match** | **no-match**} **timer** *timer-name* **Example:** `Device(config-filter-control-classmap)# match timer restart`	(Optional) Creates a condition that evaluates true based on an event's timer.
Step 18	{**match** | **no-match**} **username** *username* **Example:** `Device(config-filter-control-classmap)# match username josmiths`	(Optional) Creates a condition that evaluates true based on an event's username.
Step 19	**end** **Example:** `Device(config-filter-control-classmap)# end`	(Optional) Exits control class-map filter configuration mode and returns to privileged EXEC mode.
Step 20	**show class-map type control subscriber** {**all** | **name** *control-class-name*} **Example:** `Device# show class-map type control subscriber all`	(Optional) Displays information about Identity-Based Networking Services control classes.

Example: Control Class

The following example shows a control class that is configured with two match conditions:

```
class-map type control subscriber match-all DOT1X_NO_AGENT
 match method dot1x
 match result-type agent-not-found
```

Configuring a Control Policy

Control policies determine the actions that the system takes in response to specified events and conditions. The control policy contains one or more control policy rules that associate a control class with one or more actions. The actions that you can configure in a policy rule depend on the type of event that you specify.

Note This task includes all of the actions that you can configure in a control policy regardless of the event. All of these actions, and their corresponding steps, are optional (steps 6 through 21 below). To display the supported actions for a particular event, use the question mark (**?**) online help function.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type control subscriber** *control-policy-name*
4. **event** *event-name* [**match-all** | **match-first**]
5. *priority-number* **class** {*control-class-name* | **always**} [**do-all** | **do-until-failure** | **do-until-success**]
6. *action-number* **activate** {**policy type control subscriber** *control-policy-name* [**child** [**no-propagation** | **concurrent**] | **service-template** *template-name* [**aaa-list** *list-name*] [**precedence** *number*] [**replace-all**]}
7. *action-number* **authenticate using** {**dot1x** | **mab** | **webauth**} [**aaa** {**authc-list** *authc-list-name* | **authz-list** *authz-list-name*]} [**merge**] [**parameter-map** *map-name*] [**priority** *priority-number*] [**replace** | **replace-all**] [**retries** *number* {**retry-time** *seconds*}]
8. *action-number* **authentication-restart** *seconds*
9. *action-number* **authorize**
10. *action-number* **clear-authenticated-data-hosts-on-port**
11. *action-number* **clear-session**
12. *action-number* **deactivate** {**policy type control subscriber** *control-policy-name* | **service-template** *template-name*}
13. *action-number* **err-disable**
14. *action-number* **pause reauthentication**
15. *action-number* **protect**
16. *action-number* **replace**
17. *action-number* **restrict**
18. *action-number* **resume reauthentication**
19. *action-number* **set-timer** *timer-name seconds*
20. *action-number* **terminate** {**dot1x** | **mab** | **webauth**}
21. *action-number* **unauthorize**

22. **end**
23. **show policy-map type control subscriber** {**all** | **name** *control-policy-name*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map type control subscriber** *control-policy-name* **Example:** `Device(config)# policy-map type control POLICY_1`	Defines a control policy for subscriber sessions.
Step 4	**event** *event-name* [**match-all** | **match-first**] **Example:** `Device(config-event-control-policymap)# event session-started`	Specifies the type of event that triggers actions in a control policy if conditions are met. • **match-all** is the default behavior. • To display the available event types, use the question mark (**?**) online help function. For a complete description of event types, see the **event** command.
Step 5	*priority-number* **class** {*control-class-name* | **always**} [**do-all** | **do-until-failure** | **do-until-success**] **Example:** `Device(config-class-control-policymap)# 10 class always`	Associates a control class with one or more actions in a control policy. • A named control class must first be configured before specifying it with the *control-class-name* argument. • **do-until-failure** is the default behavior.
Step 6	*action-number* **activate** {**policy type control subscriber** *control-policy-name* [**child** [**no-propagation** | **concurrent**] | **service-template** *template-name* [**aaa-list** *list-name*] [**precedence** *number*] [**replace-all**]} **Example:** `Device(config-action-control-policymap)# 10 activate service-template FALLBACK`	(Optional) Activates a control policy or service template on a subscriber session.
Step 7	*action-number* **authenticate using** {**dot1x** | **mab** | **webauth**} [**aaa** {**authc-list** *authc-list-name* | **authz-list** *authz-list-name*}] [**merge**] [**parameter-map** *map-name*] [**priority** *priority-number*] [**replace** | **replace-all**] [**retries** *number* {**retry-time** *seconds*}] **Example:**	(Optional) Initiates the authentication of a subscriber session using the specified method.

	Command or Action	Purpose
	`Device(config-action-control-policymap)# 10 authenticate using dot1x priority 10`	
Step 8	*action-number* **authentication-restart** *seconds* **Example:** `Device(config-action-control-policymap)# 20 authentication-restart 60`	(Optional) Sets a timer to restart the authentication process after an authentication or authorization failure.
Step 9	*action-number* **authorize** **Example:** `Device(config-action-control-policymap)# 10 authorize`	(Optional) Initiates the authorization of a subscriber session.
Step 10	*action-number* **clear-authenticated-data-hosts-on-port** **Example:** `Device(config-action-control-policymap)# 20 clear-authenticated-data-hosts-on-port`	(Optional) Clears authenticated data hosts on a port after an authentication failure.
Step 11	*action-number* **clear-session** **Example:** `Device(config-action-control-policymap)# 30 clear-session`	(Optional) Clears an active subscriber session.
Step 12	*action-number* **deactivate** {**policy type control subscriber** *control-policy-name* | **service-template** *template-name*} **Example:** `Device(config-action-control-policymap)# 20 deactivate service-template`	(Optional) Deactivates a control policy or service template on a subscriber session.
Step 13	*action-number* **err-disable** **Example:** `Device(config-action-control-policymap)# 10 err-disable`	(Optional)Temporarily disables a port after a session violation event.
Step 14	*action-number* **pause reauthentication** **Example:** `Device(config-action-control-policymap)# 20 pause reauthentication`	(Optional) Pauses reauthentication after an authentication failure.
Step 15	*action-number* **protect** **Example:** `Device(config-action-control-policymap)# 10 protect`	(Optional) Silently drops violating packets after a session violation event.
Step 16	*action-number* **replace** **Example:** `Device(config-action-control-policymap)# 10 replace`	(Optional) Clears the existing session and creates a new session after a violation event.

	Command or Action	Purpose
Step 17	*action-number* **restrict** **Example:** `Device(config-action-control-policymap)# 10 restrict`	(Optional) Drops violating packets and generates a syslog entry after a session violation event.
Step 18	*action-number* **resume reauthentication** **Example:** `Device(config-action-control-policymap)# 20 resume reauthentication`	(Optional) Resumes the reauthentication process after an authentication failure.
Step 19	*action-number* **set-timer** *timer-name seconds* **Example:** `Device(config-action-control-policymap)# 20 set-timer RESTART 60`	(Optional) Starts a named policy timer.
Step 20	*action-number* **terminate** {**dot1x** | **mab** | **webauth**} **Example:** `Device(config-action-control-policymap)# 20 terminate webauth`	(Optional) Terminates an authentication method on a subscriber session.
Step 21	*action-number* **unauthorize** **Example:** `Device(config-action-control-policymap)# 20 unauthorize`	(Optional) Removes all authorization data from a subscriber session.
Step 22	**end** **Example:** `Device(config-action-control-policymap)# end`	(Optional) Exits control policy-map action configuration mode and returns to privileged EXEC mode.
Step 23	**show policy-map type control subscriber** {**all** | **name** *control-policy-name*} **Example:** `Device# show policy-map type control subscriber name POLICY_1`	(Optional) Displays information about identity control policies.

Example: Control Policy

The following example shows a simple control policy with the minimum configuration necessary for initiating authentication:

```
policy-map type control subscriber POLICY_1
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using dot1x
```

For detailed examples of control policies for concurrent and sequential authentication, see the "Configuration Examples for Identity Control Policies, on page 30" section.

Applying a Control Policy to an Interface

Control policies typically control the authentication of subscriber identity and the activation of services on sessions. Perform this task to apply a control policy to an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service-policy type control subscriber** *control-policy-name*
5. **subscriber aging** {**inactivity-timer** *seconds* [**probe**] | **probe**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/1`	Specifies an interface and enters interface configuration mode.
Step 4	**service-policy type control subscriber** *control-policy-name* **Example:** `Device(config-if)# service-policy type control subscriber POLICY_1`	Applies a previously configured control policy. • To display a list of all configured control policies, use the question mark (**?**) online help function.
Step 5	**subscriber aging** {**inactivity-timer** *seconds* [**probe**]	**probe**} **Example:** `Device(config-if)# subscriber aging inactivity-timer 60 probe`	Enables an inactivity timer for subscriber sessions.

Example: Applying a Control Policy to an Interface

```
interface GigabitEthernet 1/0/2
 subscriber aging inactivity-timer 60 probe
 service-policy type control subscriber POLICY_1
```

Configuring Authentication Features on Ports

Perform this task to control access to a port, including the port authorization state, host access mode, preauthentication access, and the authentication direction.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **access-session port-control** {**auto** | **force-authorized** | **force-unauthorized**}
5. **access-session host-mode** {**multi-auth** | **multi-domain** | **multi-host** | **single-host**}
6. **access-session closed**
7. **access-session control-direction** {**both** | **in**}
8. **end**
9. **show access-session interface** *interface-type interface-number* [**details**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/0/2`	Enters interface configuration mode for the selected interface.
Step 4	**access-session port-control** {**auto**	**force-authorized**	**force-unauthorized**} **Example:** `Device(config-if)# access-session port-control auto`	Sets the authorization state of a port. • The default value is **force-authorized**.
Step 5	**access-session host-mode** {**multi-auth**	**multi-domain**	**multi-host**	**single-host**} **Example:** `Device(config-if)# access-session host-mode single-host`	Allows hosts to gain access to a controlled port. • To use this command, you must first enable the **access-session port-control auto** command. • The default value is **multi-auth**.
Step 6	**access-session closed** **Example:** `Device(config-if)# access-session closed`	Prevents preauthentication access on this port. • The port is set to open access by default.

	Command or Action	Purpose
Step 7	**access-session control-direction** {**both** | **in**} **Example:** `Device(config-if)# access-session control-direction in`	Sets the direction of authentication control on a port. • The default value is **both**.
Step 8	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 9	**show access-session interface** *interface-type interface-number* [**details**] **Example:** `Device# show access-session interface gigabitethernet 1/0/2 details`	Displays information about subscriber sessions that match the specified client interface.

Example: Port Authentication

```
interface GigabitEthernet 1/0/2
 access-session host-mode single-host
 access-session closed
 access-session port-control auto
 access-session control-direction in
```

Configuring a Parameter Map for Web-Based Authentication

A parameter map allows you to modify parameters that control the behavior of actions configured under a control policy. A parameter map for web-based authentication sets parameters that can be applied to subscriber sessions during authentication. If you do not create a parameter map, the policy uses default parameters.

Perform the following steps to define either a global or named parameter map for web-based authentication.

Note The configuration commands available in the global parameter map differ from the commands available in a named parameter map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type webauth** {*parameter-map-name* | **global**}
4. **banner** {**file** *location*:*filename* | **text** *banner-text*}
5. **consent**
6. **consent email**
7. **custom-page** {**failure** | **login [expired]** | **success**} **device** *location*:*filename*
8. **max-http-conns** *number*

9. **ratelimit init-state-sessions** *rate-limit*
10. **redirect** {{**for-login** | **on-failure** | **on-success**} *url* | **portal** {**ipv4** *ipv4-address* | **ipv6** *ipv6-address*}}
11. **timeout init-state sec** *seconds*
12. **type** {**authbypass** | **consent** | **webauth** | **webconsent**}
13. **virtual-ip** {**ipv4** *ipv4-address* | **ipv6** *ipv6-address*}
14. **watch-list** {**add-item** {**ipv4** *ipv4-address* | **ipv6** *ipv6-address*} | **dynamic-expiry-timeout** *minutes* | **enabled**}
15. **end**
16. **show ip admission status** [**banners** | **custom-pages** | **parameter-map** [*parameter-map*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type webauth** {*parameter-map-name*	**global**} **Example:** `Device(config)# parameter-map type webauth MAP_2`	Creates a parameter map and enters parameter-map webauth configuration mode. • The specific configuration commands supported for a global parameter map defined with the **global** keyword differ from the commands supported for a named parameter map defined with the *parameter-map-name* argument.
Step 4	**banner** {**file** *location*:*filename*	**text** *banner-text*} **Example:** `Device(config-params-parameter-map)# banner file flash:webauth_banner.html`	(Optional) Displays a banner on the web-authentication login web page.
Step 5	**consent** **Example:** `Device(config-params-parameter-map)# type consent`	(Optional) Defines the methods supported by a web-based authentication parameter map. • This command is supported in named parameter maps only.
Step 6	**consent email** **Example:** `Device(config-params-parameter-map)# consent email`	(Optional) Requests a user's e-mail address on the web-authentication login web page. • This command is supported in named parameter maps only.
Step 7	**custom-page** {**failure**	**login [expired]**	**success**} **device** *location*:*filename*	(Optional) Displays custom authentication proxy web pages during web-based authentication.

	Command or Action	Purpose

	Example:
	`Device(config-params-parameter-map)# custom-page login device flash:webauth_login.html`
	`Device(config-params-parameter-map)# custom-page login expired device flash:webauth_expire.html`
	`Device(config-params-parameter-map)# custom-page success device flash:webauth_success.html`
	`Device(config-params-parameter-map)# custom-page failure device flash:webauth_fail.html`
		• You must configure all four custom HTML files. If fewer than four files are configured, the internal default HTML pages will be used.

Step 8	**max-http-conns** *number* **Example:** `Device(config-params-parameter-map)# max-http-conns 5`	(Optional) Limits the number of HTTP connections for each web authentication client.
Step 9	**ratelimit init-state-sessions** *rate-limit* **Example:** `Device(config-params-parameter-map)# ratelimit init-state-sessions 500`	(Optional) Limits the number of web-based authentication sessions in the Init state. • This command is supported in the global parameter map only.

Step 10	**redirect** {{**for-login** | **on-failure** | **on-success**} *url* | **portal** {**ipv4** *ipv4-address* | **ipv6** *ipv6-address*}}
	Example:
	`Device(config-params-parameter-map)# redirect portal ipv6 FE80::1`
	`Device(config-params-parameter-map)# redirect on-failure http://10.10.3.34/~sample/failure.html`
		(Optional) Redirects users to a particular URL during web-based authentication.

Step 11	**timeout init-state sec** *seconds* **Example:** `Device(config-params-parameter-map)# timeout init-state sec 60`	(Optional) Sets the Init state timeout for web-based authentication sessions. • The range of seconds is (60-3932100).
Step 12	**type** {**authbypass** | **consent** | **webauth** | **webconsent**} **Example:** `Device(config-params-parameter-map)# type consent`	(Optional) Defines the methods supported by a web-based authentication parameter map. • This command is supported in named parameter maps only.
Step 13	**virtual-ip** {**ipv4** *ipv4-address* | **ipv6** *ipv6-address*} **Example:** `Device(config-params-parameter-map)# virtual-ip ipv6 FE80::1`	(Optional) Specifies a virtual IP address for web-based authentication clients. • This command is supported in the global parameter map only.
Step 14	**watch-list** {**add-item** {**ipv4** *ipv4-address* | **ipv6** *ipv6-address*} | **dynamic-expiry-timeout** *minutes* | **enabled**} **Example:**	(Optional) Enables a watch list of web-based authentication clients. • This command is supported in the global parameter map only.

	Command or Action	Purpose

	`Device(config-params-parameter-map)# watch-list enabled`
	`Device(config-params-parameter-map)# watch-list dynamic-expiry-timeout 20`
	`Device(config-params-parameter-map)# watch-list add-item ipv6 FE80::1`	

Step 15	**end** **Example:** `Device(config-params-parameter-map)# end`	(Optional) Exits parameter-map configuration mode and returns to privileged EXEC mode.
Step 16	**show ip admission status** [**banners** | **custom-pages** | **parameter-map** [*parameter-map*]] **Example:** `Device# show ip admission status custom-pages`	(Optional) Displays information about configured banners and custom pages.

Example: Parameter Map for Web-Based Authentication

```
parameter-map type webauth PMAP_2
 type webconsent
 timeout init-state sec 60
 max-http-conns 5
 type consent
 consent email
 custom-page login device flash:webauth_login.html
 custom-page success device flash:webauth_success.html
 custom-page failure device flash:webauth_fail.html
 custom-page login expired device flash:webauth_expire.html
```

What to do next

Apply the parameter map to sessions by specifying it in the **authenticate using** command when configuring a Control Policy. See the "Configuring a Control Policy, on page 21" section.

Configuration Examples for Identity Control Policies

Example: Configuring Control Policy for Concurrent Authentication Methods

The following example shows a control policy that is configured to allow concurrent authentication. All three methods (dot1x, MAB, and web authentication) are run simultaneously when a session is started. The dot1x method is set to the highest priority and web authentication has the lowest priority, which means that if multiple methods succeed, the highest priority method is honored.

If authentication fails, the session manager checks whether all methods have failed, and if so, it sets the restart timer to 60 seconds, after which it attempts to start all three methods again. On authentication success, the session manager terminates any lower priority methods; for dot1x, this is MAB and webauth; for MAB it is webauth. Lastly, if session manager detects a dot1x client (agent-found) it triggers only dot1x to run.

The class map named ALL-FAILED checks that all three methods have run to completion (result type is none until then) and that none of them was successful. In other words, all three methods have completed and failed.

Note When configuring a control policy for concurrent authentication, you must include a policy rule that explicitly terminates one method after another method of a higher priority succeeds.

```
class-map type subscriber control match-all ALL_FAILED
 no-match result-type method dot1x none
 no-match result-type method dot1x success
 no-match result-type method mab none
 no-match result-type method mab success
 no-match result-type method webauth none
 no-match result-type method webauth success
!
class-map type control subscriber match-all DOT1X
 match method dot1x
!
class-map type control subscriber match-all MAB
 match method mab
!
policy-map type control subscriber CONCURRENT_DOT1X_MAB_WEBAUTH
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using mab priority 20
   20 authenticate using dot1x priority 10
   30 authenticate using webauth parameter-map WEBAUTH_DEFAULT priority 30
 event authentication-failure match-first
  10 class ALL_FAILED
   10 authentication-restart 60
 event authentication-success match-all
  10 class DOT1X
   10 terminate MAB
   20 terminate webauth
  20 class MAB
   10 terminate webauth
 event agent-found match-all
  10 class always do-until-failure
   10 authenticate using dot1x priority 10
```

Example: Configuring Control Policy for Sequential Authentication Methods

The following example shows a control policy that is configured to allow sequential authentication methods using 802.1X (dot1x), MAB, and web authentication.

```
parameter-map type webauth WEBAUTH_FALLBACK
 type webauth
!
class-map type control subscriber match-all DOT1X_NO_RESP
 match method dot1x
 match result-type method dot1x agent-not-found
!
class-map type control subscriber match-all MAB_FAILED
 match method mab
 match result-type method mab authoritative
!
```

```
policy-map type control subscriber POLICY_Gi3/0/10
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using dot1x priority 10
 event authentication-failure match-first
  10 class DOT1X_NO_RESP do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  20 class MAB_FAILED do-until-failure
   10 terminate mab
   20 authenticate using webauth parameter-map WEBAUTH_FALLBACK priority 30
  30 class always do-until-failure
   10 terminate dot1x
   20 terminate mab
   30 terminate webauth
   40 authentication-restart 60
 event agent-found match-all
  10 class always do-until-failure
   10 terminate mab
   20 terminate webauth
   30 authenticate using dot1x priority 10
```

The following example shows a control policy that is configured to allow sequential authentication methods using 802.1X and MAB. If authentication fails, a service template for VLAN is activated.

```
service-template VLAN210
 vlan 210
!
class-map type control subscriber match-all DOT1X_FAILED
 match method dot1x
 match result-type method dot1x authoritative
!
class-map type control subscriber match-all DOT1X_NO_RESP
 match method dot1x
 match result-type method dot1x agent-not-found
!
class-map type control subscriber match-all MAB_FAILED
 match method mab
 match result-type method mab authoritative
!
policy-map type control subscriber POLICY_Gi3/0/14
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using dot1x retries 2 retry-time 0 priority 10
 event authentication-failure match-first
  10 class DOT1X_NO_RESP do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  20 class MAB_FAILED do-until-failure
   10 terminate mab
   20 activate service-template VLAN210
   30 authorize
  30 class DOT1X_FAILED do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  40 class always do-until-failure
   10 terminate dot1x
   20 terminate mab
   30 authentication-restart 60
 event agent-found match-all
  10 class always do-until-failure
   10 terminate mab
   20 authenticate using dot1x retries 2 retry-time 0 priority 10
```

Example: Configuring Parameter Maps

Global Parameter Map

The following example shows the configuration of a global parameter map:

```
parameter-map type webauth global
 timeout init-state min 15
 logging enabled
 watch-list enabled
 virtual-ip ipv6 FE80::1
 redirect on-failure http://10.10.3.34/~sample/failure.html
 ratelimit init-state-sessions 500
 max-http-conns 100
 watch-list dynamic-expiry-timeout 5000
 banner file flash:webauth_banner.html
```

Named Parameter Maps for Web Authentication and Authentication Bypass (nonresponsive host [NRH])

The following example shows the configuration of two named parameter maps; one for web authentication and one for authentication bypass. This example also shows the corresponding control policy configuration.

```
parameter-map type webauth WEBAUTH_BANNER
 type webauth
 banner
!
parameter-map type webauth WEBAUTH_NRH
 type authbypass
!
class-map type control subscriber match-all NRH_FAIL
 match method webauth
 match current-method-priority eq 254
!
policy-map type control subscriber WEBAUTH_NRH
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using webauth parameter-map WEBAUTH_NRH priority 254
 event authentication-failure match-all
  10 class NRH_FAIL do-until-failure
   10 terminate webauth
   20 authenticate using webauth parameter-map WEBAUTH_BANNER priority 30
```

Named Parameter Map for Web Authentication Using Custom Pages

The following example shows the configuration of a named parameter map for web authentication that defines custom pages for the login process, along with a control policy that uses the parameter map.

```
parameter-map type webauth CUSTOM_WEBAUTH
 type webauth
 custom-page login device flash:login_page.htm
 custom-page success device flash:success_page.htm
 custom-page failure device flash:fail_page.htm
 custom-page login expired device flash:expire_page.htm
!
policy-map type control subscriber CUSTOM_WEBAUTH
```

```
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using webauth parameter-map CUSTOM_WEB retries 2 retry-time 0
```

Named Parameter Map for Consent

The following example shows the configuration of a named parameter map for consent, along with the corresponding control policy that uses the parameter map:

```
parameter-map type webauth CONSENT
 type consent
!
ip access-list extended GUEST_ACL
 permit ip any 172.30.30.0 0.0.0.255
 permit ip any host 172.20.249.252
!
service-template GUEST_POLICY
 access-group GUEST_ACL
!
policy-map type control subscriber CONSENT
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using webauth parameter-map CONSENT
 event authentication-success match-all
  10 class always do-until-failure
   10 activate service-template GUEST_POLICY
```

Named Parameter Map for Web Authentication with Consent

The following example shows the configuration of a named parameter map for web authentication with consent, along with the corresponding control policy that uses the parameter map:

```
parameter-map type webauth WEBAUTH_CONSENT
 type webconsent
!
ip access-list extended GUEST_ACL
 permit ip any 172.30.30.0 0.0.0.255
 permit ip any host 172.20.249.252
!
service-template GUEST_POLICY
 access-group GUEST_ACL
!
policy-map type control subscriber WEBAUTH_CONSENT
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using webauth parameter-map CONSENT
 event authentication-success match-all
  10 class always do-until-failure
   10 activate service-template GUEST_POLICY
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Identity-Based Networking Services commands	Cisco IOS Identity-Based Networking Services Command Reference
Address Resolution Protocol (ARP) commands	Cisco IOS IP Addressing Services Command Reference
ARP configuration tasks	IP Addressing - ARP Configuration Guide
Authentication, authorization, and accounting (AAA) configuration tasks	Authentication Authorization and Accounting Configuration Guide
AAA commands	Cisco IOS Security Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 5176	Dynamic Authorization Extensions to RADIUS

Technical Assistance

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

Feature Information for Identity Control Policies

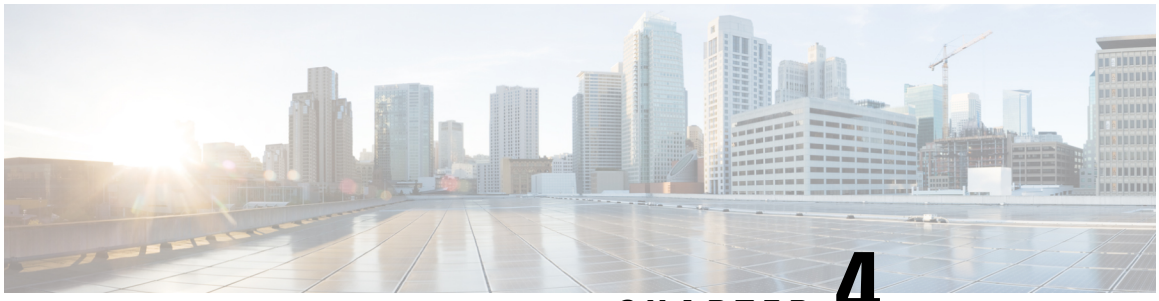
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train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

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

Table 4: Feature Information for Identity Control Policies

Feature Name	Releases	Feature Information
Cisco Common Classification Policy Language-Based Identity Configuration	Cisco IOS XE Release 3.2SE	Identity control policies define the actions taken in response to specified events and conditions. In Cisco IOS XE 3.2SE, this feature is supported on the following platforms: • Cisco Catalyst 3650 Series Switches • Cisco Catalyst 3850 Series Switches • Cisco Catalyst 4500E Supervisor Engine 6-E • Cisco Catalyst 4500E Supervisor Engine 6L-E • Cisco Catalyst 4500E Supervisor Engine 7-E • Cisco Catalyst 4500 E Supervisor Engine 7L-E The following commands were introduced: **activate (policy-map action)**, **authenticate using**, **authentication display**, **authentication-restart**, **authorize**, **banner (parameter-map webauth)**, **class**, **class-map type control subscriber**, **clear-authenticated-data-hosts-on-port**, **clear session**, **consent email custom-page**, **deactivate**, **err-disable**, **event**, **logging enabled (parameter-map webauth)**, **match**, **max-http-conns**, **parameter-map type webauth**, **pause reauthentication**, **policy-map type control subscriber**, **protect (policy-map action)**, **ratelimit init-state-sessions**, **redirect (parameter-map webauth)**, **replace**, **restrict**, **resume reauthentication**, **service-policy type control subscriber**, **set-timer**, **show access-session**, **show class-map type control subscriber**, **show policy-map type control subscriber**, **terminate**, **type (parameter-map webauth)**, **unauthorize**, **virtual-ip**, **watch-list**.

Feature Name	Releases	Feature Information
Concurrent Authentication	Cisco IOS XE Release 3.2SE	Allows concurrent operation of 802.1x, MAB and web authentication methods, making it possible to invoke multiple authentication methods in parallel on a single session. In Cisco IOS XE 3.2SE, this feature is supported on the following platforms: • Cisco Catalyst 3650 Series Switches • Cisco Catalyst 3850 Series Switches • Cisco Catalyst 4500E Supervisor Engine 6-E • Cisco Catalyst 4500E Supervisor Engine 6L-E • Cisco Catalyst 4500E Supervisor Engine 7-E • Cisco Catalyst 4500E Supervisor Engine 7L-E
Per User Inactivity Handling across Methods	Cisco IOS XE Release 3.2SE	Supports RADIUS attributes 28 (Idle-Timeout) and 29 (Termination-Action). In Cisco IOS XE 3.2SE, this feature is supported on the following platforms: • Cisco Catalyst 3650 Series Switches • Cisco Catalyst 3850 Series Switches • Cisco Catalyst 4500E Supervisor Engine 6-E • Cisco Catalyst 4500E Supervisor Engine 6L-E • Cisco Catalyst 4500E Supervisor Engine 7-E • Cisco Catalyst 4500E Supervisor Engine 7L-E



CHAPTER 4

Policy Classification Engine

The Policy Classification Engine feature helps configure device-based policies and client (network endpoint) profiling and enforces a per user or per device policy on a network. The policy classification engine enables bring-your-own-device (BYOD) deployments integrate user or wireless device policies into the wireless controller. This module explains how to configure policies and apply them to a wireless LAN (WLAN).

Finding Feature Information

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

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

Restrictions for Policy Classification Engine

Interface templates are not valid on wireless sessions.

Information About Policy Classification Engine

Policy Classification Engine Overview

The Policy Classification Engine feature helps configure device-based policies and client (network endpoint) profiling and enforces a per user or per device policy on a network.

You can configure sets of different policies that can be used for lookup and sequential matching. A policy is matched based on the configured policy statement. Use policies to profile devices based on the Dynamic Host Control Protocol (DHCP) or HTTP to identify end devices in a network. You can enforce specific policies at network endpoints.

The device (switch; for example, Cisco Catalyst 3850 Wireless LAN Controller) uses these attributes and predefined classification profiles to identify devices.

Policies are configured based on the following parameters:

- Device—Types of end devices. Examples are Windows machines, smart phones, Apple device like iPads, iPhones, and so on.
- Regular expressions
- User role—The user type or user group to which an user belongs. Examples are students, employees, and so on.
- Username—Login credentials entered by users.
- Time-of-day—The time-of-day when endpoints are allowed into a network.
- OUI—The MAC address that identifies the Organizational Unique Identifier (OUI).
- MAC address—The MAC address of the endpoint.

Once the device (switch) has a match corresponding to the policy parameters per end point, a policy is added. Policy enforcement is based on the following session attributes:

- VLAN—User-defined VLAN
- Access control list (ACL)
- Session timeout value—User-defined timeout for client sessions
- Quality of service (QoS)

You can configure policies and based on the session attributes, enforce these policies on end points.

How to Configure Policy Classification Engine

Configuring Policies in Identity-Based Networking Services

To configure policies, perform the following tasks:

1. Configure a service template.

   For more information, see the Configuring Identity Services Templates module.

2. Configure an interface template.

   For more information, see the Interface Templates module.
3. Create a parameter map.
4. Create a policy map.
5. Apply the policy on a wireless LAN (WLAN).

Configuring a Subscriber Parameter Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type subscriber attribute-to-service** *parameter-map-name*
4. *priority-number* **map device-type eq** *device-type* **oui eq** *MAC-address*
5. *action-number* **interface-template** *interface-template-name*
6. **end**
7. **show parameter-map type subscriber attribute-to-service** *parameter-map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type subscriber attribute-to-service** *parameter-map-name* **Example:** `Device(config)# parameter-map type subscriber attribute-to-service param-map`	Configures a subscriber parameter map and enters parameter-map filter configuration mode.
Step 4	*priority-number* **map device-type eq** *device-type* **oui eq** *MAC-address* **Example:** `Device(config-parameter-map-filter)# 1 map device-type eq "Cisco-IP-Phone-9971" oui "eq 08.cc.68"`	Maps the priority and the Organizationally Unique Identifier (OUI) of the configured device, and enters parameter-map filter submode configuration mode.
Step 5	*action-number* **interface-template** *interface-template-name* **Example:**	Maps the action number to an interface template.

	Command or Action	Purpose
	`Device(config-parameter-map-filter-submode)# 2 interface-template IP-PHONE-INTERFACE-TEMPLATE`	
Step 6	**end** **Example:** `Device(config-parameter-map-filter-submode)# end`	Exits parameter-map filter submode configuration mode and returns to privileged EXEC mode.
Step 7	**show parameter-map type subscriber attribute-to-service** *parameter-map-name* **Example:** `Device# parameter-map type subscriber attribute-to-service parameter-map-name`	Displays information about the specified parameter map.

Example

The following is sample output from the **show parameter-map type subscriber attribute-to-service** command:

```
Device# show parameter-map type subscriber attribute-to-service param-map

Parameter-map name: param-map
 Map: 1 map device-type eq "Cisco-IP-Phone-9971" oui eq "08.cc.68"
  Action(s):
   2 interface-template IP-PHONE-INTERFACE-TEMPLATE
```

Configuring a Subscriber Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type control subscriber** *policy-map-name*
4. **event identity-update** {**match-all** | **match-first**}
5. *priority-number* **class always** {**do-all** | **do-until-failure** | **do-until-success**}
6. *action-number* **map attribute-to-service table** *parameter-map-name*
7. **end**
8. **show policy-map type control subscriber** *policy-map-name*

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map type control subscriber** *policy-map-name* **Example:** `Device(config)# policy-map type control subscriber pmap`	Defines a control policy for subscriber sessions and enters control policy-map event configuration mode.
Step 4	**event identity-update** {**match-all** | **match-first**} **Example:** `Device(config-event-control-policymap)# event identity-update match-all`	Specifies the event type that triggers actions in a control policy if conditions are met, and enters control policy-map class configuration mode.
Step 5	*priority-number* **class always** {**do-all** | **do-until-failure** | **do-until-success**} **Example:** `Device(config-class-control-policymap)# 1 class always do-until-failure`	Associates a control class with one or more actions in a control policy and enters control policy-map action configuration mode.
Step 6	*action-number* **map attribute-to-service table** *parameter-map-name* **Example:** `Device(config-action-control-policymap)# 2 map attribute-to-service table param-map`	Maps identity-update attribute to an autoconf template.
Step 7	**end** **Example:** `Device(config-action-control-policymap)# end`	Exits control policy-map action configuration mode and returns to privileged EXEC mode.
Step 8	**show policy-map type control subscriber** *policy-map-name* **Example:** `Device# show policy-map type control subscriber pmap`	Displays information and statistics about the control policies.

Example

The following is sample output from the **show policy-map type control subscriber** command:

```
Device# show policy-map type control subscriber pmap

show policy-map type control subscriber pmap
policy-map
  event identity-update match-all
    1 class always do-until-failure
      1 map attribute-to-service table param-map
```

Applying a Subscriber Policy to a WLAN

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **wlan** *wlan-name wlan-ID SSID*
4. **service-policy type control subscriber** *policy-map-name*
5. **profiling local http**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**wlan** *wlan-name wlan-ID SSID* **Example:** `Device(config)# wlan wlan1 9 policywlan`	Configures a wireless LAN (WLAN) network and enters WLAN configuration mode.
Step 4	**service-policy type control subscriber** *policy-map-name* **Example:** `Device(config-wlan)# service-policy type control subscriber pmap`	Defines a service policy for subscriber sessions.
Step 5	**profiling local http** **Example:** `Device(config-wlan)# profiling local http`	Configures client profiling on a WLAN based on HTTP attributes.
Step 6	**end** **Example:** `Device(config-wlan)# end`	Exits WLAN configuration mode and returns to privileged EXEC mode.

Configuration Examples for Policy Classification Engine

Example: Configuring a Subscriber Parameter Map

```
Device# configure terminal
Device(config)# parameter-map type subscriber attribute-to-service param-map
Device(config-parameter-map-filter)# 1 map device-type eq "Cisco-IP-Phone-9971" oui "eq
08.cc.68"
Device(config-parameter-map-filter-submode)# 2 interface-template IP-PHONE-INTERFACE-TEMPLATE
Device(config-parameter-map-filter-submode)# end
```

Example: Configuring a Subscriber Policy Map

```
Device# configure terminal
Device(config)# policy-map type control subscriber pmap
Device(config-event-control-policymap)# event identity-update match-all
Device(config-class-control-policymap)# 1 class always do-until-failure
Device(config-action-control-policymap)# 2 map attribute-to-service table param-map
Device(config-action-control-policymap)# end
```

Example: Applying a Subscriber Policy to a WLAN

```
Device# configure terminal
Device(config)# wlan wlan1 9 policywlan
Device(config-wlan)# service-policy type control subscriber pmap
Device(config-wlan)# profiling local http
Device(config-wlan)# end
```

Additional References for Policy Classification Engine

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Identity commands	Cisco IOS Identity-Based Networking Services Command Reference
Service templates	"Configuring Identity Service Templates" module of the `Identity-Based Networking Services Configuration Guide`
Interface templates	"Interface Templates" module of the `Identity-Based Networking Services Configuration Guide`

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

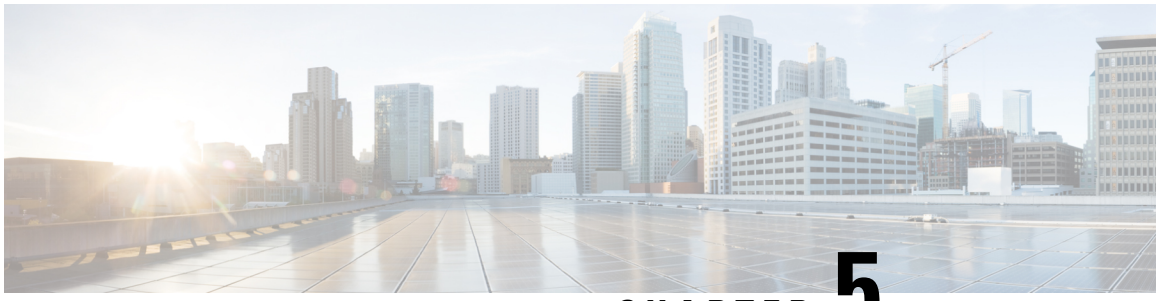
Feature Information for Policy Classification Engine

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

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

Table 5: Feature Information for Policy Classification Engine

Feature Name	Releases	Feature Information
Policy Classification Engine	Cisco IOS XE Release 3.6E	The Policy Classification Engine feature helps configure device-based policies and client (network endpoint) profiling and enforces a per user or per device policy on a network. The policy classification engine enables bring-your-own-device (BYOD) deployments integrate user or wireless device policies into the wireless controller. In Cisco IOS XE 3.6E, this feature is supported on the following platforms: • Cisco 5700 Series Wireless LAN Controllers • Cisco Catalyst 3650 Series Switches • Cisco Catalyst 3850 Series Switches



CHAPTER 5

Configuring Identity Service Templates

Identity service templates contain a set of policy attributes or features that can be applied to one or more subscriber sessions through a control policy, a RADIUS Change of Authorization (CoA) request, or a user profile or service profile. This module provides information about how to configure local service templates for Identity-Based Networking Services.

Finding Feature Information

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

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

Prerequisites for Identity Service Templates

For downloadable service templates, the switch uses the default password “cisco123” when downloading the service templates from the authentication, authorization, and accounting (AAA) server, Cisco Secure Access Control Server (ACS), or Cisco Identity Services Engine (ISE). The AAA, ACS, and ISE server must include the password “cisco123” in the service template configuration.

Information About Identity Service Templates

Service Templates for Identity-Based Networking Services

A service template contains a set of service-related attributes or features, such as access control lists (ACLs) and VLAN assignments, that can be activated on one or more subscriber sessions in response to session life-cycle events. Templates simplify the provisioning and maintenance of network session policies where policies fall into distinct groups or are role-based.

A service template is applied to sessions through its reference in a control policy, through RADIUS Change of Authorization (CoA) requests, or through a user profile or service profile. User profiles are defined per subscriber; service profiles can apply to multiple subscribers.

Identity-Based Networking Services supports two types of service templates:

- Downloadable Service Templates—The service template is configured centrally on an external ACS or AAA server and downloaded on demand.
- Locally Configured Service Templates—The service template is configured locally on the device through the Cisco IOS command-line interface (CLI).

Downloadable Service Templates

Identity-Based Networking Services can download a service template defined on an external AAA server. The template defines a collection of AAA attributes. These templates are applied to sessions through the use of vendor-specific attributes (VSAs) included in RADIUS CoA messages received from the external AAA server or ACS. The name of the template is referenced in a user profile or a control policy, which triggers a download of the service template during processing.

The downloadable template is cached on the device and subsequent requests for a download will refer to the available cached template. The template however is cached only for the duration of its active usage. The downloaded template cached on the device is protected and cannot be deleted through the command line interface or through other applications. This ensures that the template is deleted only when there are no active references to it.

Locally Configured Service Templates

Service templates can be configured locally through the CLI. These service templates can be applied to subscriber sessions by a reference in a control policy.

When an active local template is updated, changes to that local template will be reflected across all sessions for which the template is active. If a template is deleted, all content from that template that is applied against sessions is removed.

How to Configure Identity Service Templates

Configuring a Local Service Template

A service template defines the local policies that can be applied to a subscriber session. Activate this service template on sessions on which the local policies must be applied.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **service-template** *template-name*
4. **absolute-timer** *minutes*
5. **access-group** *access-list-name*
6. **description** *description*
7. **inactivity-timer** *minutes* **[probe]**
8. **redirect url** *url*
9. **sgt** *range*
10. **tag** *tag-name*
11. **vlan** *vlan-id*
12. **sgt** *sgt-tag*
13. **end**
14. **show service-template** [*template-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**service-template** *template-name* **Example:** `Device(config)# service-template SVC_2`	Creates a service template and enters service template configuration mode.
Step 4	**absolute-timer** *minutes* **Example:** `Device(config-service-template)# absolute-timer 15`	(Optional) Enables an absolute timeout for subscriber sessions.

	Command or Action	Purpose
Step 5	**access-group** *access-list-name* **Example:** `Device(config-service-template)# access-group ACL_2`	(Optional) Applies an access list to sessions using a service template.
Step 6	**description** *description* **Example:** `Device(config-service-template)# description label for SVC_2`	(Optional) Adds a description for a service template.
Step 7	**inactivity-timer** *minutes* **[probe]** **Example:** `Device(config-service-template)# inactivity-timer 15`	(Optional) Enables an inactivity timeout for subscriber sessions.
Step 8	**redirect url** *url* **Example:** `Device(config-service-template)# redirect url www.cisco.com`	(Optional) Redirects clients to a particular URL.
Step 9	**sgt** *range* **Example:** `Device(config-service-template)# sgt 100`	(Optional) Associates a Security Group Tag (SGT) with a service template.
Step 10	**tag** *tag-name* **Example:** `Device(config-service-template)# tag TAG_2`	(Optional) Associates a user-defined tag with a service template.
Step 11	**vlan** *vlan-id* **Example:** `Device(config-service-template)# vlan 215`	(Optional) Applies a VLAN to sessions using a service template.
Step 12	**sgt** *sgt-tag* **Example:** `Device(config-service-template)# sgt`	(Optional) Adds a Security Group Tag (SGT) using a service template.
Step 13	**end** **Example:** `Device(config-service-template)# end`	Exits service template configuration mode and returns to privileged EXEC mode.
Step 14	**show service-template** [*template-name*] **Example:** `Device# show service-template SVC_2`	Displays information about configured service templates.

Example: Service Template

```
service-template SVC_2
 description label for SVC_2
 access-group ACL_2
 redirect url www.cisco.com
 vlan 215
 inactivity-timer 15
 absolute-timer 15
 tag TAG_2
```

What to do next

To activate a service template on a subscriber session, specify the service template in a control policy. See "Configuring a Control Policy, on page 21."

Configuration Examples for Identity Service Templates

Example: Activating a Service Template and Replace All

Local Service Template Configuration

The following example shows the configuration of a service template defined locally on the device. This template contains attributes that are applied to sessions that use the control policy named POSTURE_VALIDATION, shown below:

```
service-template DOT1X
 access-group SVC1_ACL
 redirect url www.cisco.com match URL_REDIRECT_ACL
 inactivity-timer 60
 absolute-timer 300
!
ip access-list extended URL_REDIRECT_ACL
 permit tcp any host 5.5.5.5 eq www
```

Control Policy Configuration

The following example shows a control policy that activates the service template named DOT1X with replace-all enabled. The successfully activated template will replace the existing authorization data and any service template previously applied to the session.

```
policy-map type control subscriber POSTURE_VALIDATION
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using dot1x priority 10
   20 authenticate using webauth priority 20
 event authentication-success match-all
  10 class DOT1X do-all
   10 terminate webauth
   20 activate service-template DOT1X replace-all
```

Example: Activating a Service Template for Fallback Service

Local Service Template Configuration

The following example shows the configuration of a service template defined locally on the device. This template contains attributes that are applied to sessions that use the control policy named POSTURE_VALIDATION, shown below:

```
service-template FALLBACK
 description fallback service
 access-group ACL_2
 redirect url www.cisco.com
 inactivity-timer 15
 absolute-timer 15
 tag TAG_2
```

Control Policy Configuration

The following example shows a control policy that runs authentication methods dot1x and MAB. If dot1x authentication fails, MAB authentication is attempted. If MAB fails, the system provides a default authorization profile using the FALLBACK template.

```
policy-map type control subscriber POSTURE_VALIDATION
 event session-started match-all
  10 class always do-all
   10 authenticate using dot1x
 event authentication-failure match-all
  10 class DOT1X do-all
   10 authenticate using mab
  20 class MAB do-all
   10 activate service-template FALLBACK
```

Example: Deactivating a Service Template

Access Control List Configuration

The following example shows the configuration of an access control list (ACL) that is used by the local service template named LOW_IMPACT_TEMPLATE, shown below.

```
ip access-list extended LOW_IMPACT_ACL
 permit udp any any eq bootps
 permit tcp any any eq www
 permit tcp any any eq 443
 permit ip any 172.30.0.0 0.0.255.255
```

Local Service Template Configuration

The following example shows the configuration of the local service template that provides limited access to all hosts even when authentication fails.

```
service-template LOW_IMPACT_TEMPLATE
 description Service template for Low impact mode
 access-group LOW_IMPACT_ACL
```

```
 inactivity-timer 60
 tag LOW_IMPACT_TEMPLATE
```

Control Policy Configuration

The following example shows the configuration of a control policy that uses the template named LOW_IMPACT_TEMPLATE to provide limited access to all hosts even when authentication fails. If authentication succeeds, the policy manager removes the service template and provides access based on the policies downloaded by the RADIUS server.

```
class-map type control subscriber match-all DOT1X_MAB_FAILED
 no-match result-type method dot1x success
 no-match result-type method mab success
!
policy-map type control subscriber CONCURRENT_DOT1X_MAB_LOW_IMP_MODE
 event session-started match-all
  10 class always do-until-failure
   10 authorize
   20 activate service-template LOW_IMPACT_TEMPLATE
   30 authenticate using mab
   40 authenticate using dot1x
 event authentication-success match-all
  10 class always do-until-failure
   10 deactivate service-template LOW_IMPACT_TEMPLATE
 event authentication-failure match-first
  10 class DOT1X_MAB_FAILED do-until-failure
   10 authorize
   20 terminate dot1x
   30 terminate mab
 event agent-found match-all
  10 class always do-until-failure
   10 authenticate using dot1x
 event inactivity-timeout match-all
  10 class always do-until-failure
   10 clear-session
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Identity-Based Networking Services commands	Cisco IOS Identity-Based Networking Services Command Reference
Address Resolution Protocol (ARP) commands	Cisco IOS IP Addressing Services Command Reference
ARP configuration tasks	IP Addressing - ARP Configuration Guide
Authentication, authorization, and accounting (AAA) configuration tasks	Authentication Authorization and Accounting Configuration Guide

Related Topic	Document Title
AAA commands	Cisco IOS Security Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 5176	Dynamic Authorization Extensions to RADIUS

Technical Assistance

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

Feature Information for Identity Service Templates

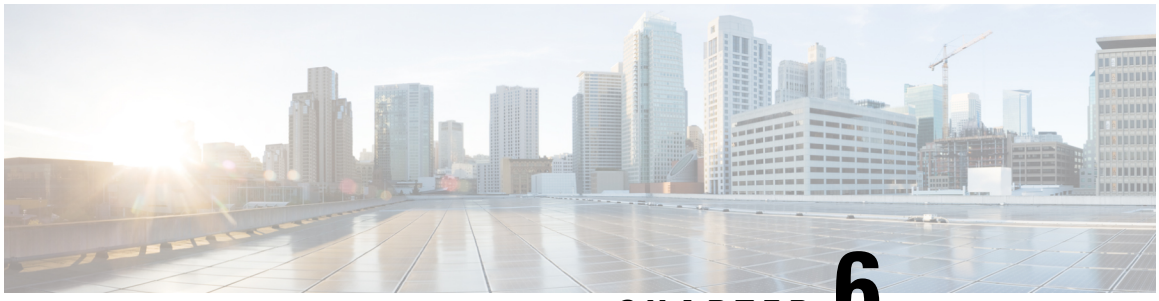
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train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

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

Table 6: Feature Information for Identity Service Templates

Feature Name	Releases	Feature Information
Downloadable Identity Service Template	Cisco IOS XE Release 3.2SE	Enables a service template to be downloaded from an ACS and its attributes applied against a session. In Cisco IOS XE 3.2SE, this feature is supported on the following platforms: • Cisco Catalyst 3650 Series Switches • Cisco Catalyst 3850 Series Switches

Feature Name	Releases	Feature Information
Identity Service Template	Cisco IOS XE Release 3.2SE	Enables identity service templates to be configured locally and available at all times. In Cisco IOS XE 3.2SE, this feature is supported on the following platforms: • Cisco Catalyst 3650 Series Switches • Cisco Catalyst 3850 Series Switches • Cisco Catalyst 4500E Supervisor Engine 6-E • Cisco Catalyst 4500E Supervisor Engine 6L-E • Cisco Catalyst 4500E Supervisor Engine 7-E • Cisco Catalyst 4500E Supervisor Engine 7L-E The following commands were introduced: **absolute-timer**, **access-group (service template)**, **description (service template)**, **inactivity-timer**, **redirect url**, **service-template**, **show service-template**, **tag (service template)**, **vlan (service template)**.



CHAPTER 6

Interface Templates

An interface template provides a mechanism to configure multiple commands at the same time and associate it with a target such as an interface. An interface template is a container of configurations or policies that can be applied to specific ports.

Finding Feature Information

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

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

Restrictions for Interface Templates

- Interface templates are not applicable for wireless sessions.
- Remote storing and downloading of templates is not supported.
- Port channel configuration through template is not supported on Cisco Catalyst 6500 Series Switches.
- On Catalyst 3650 Series Switches and Catalyst 3850 Series Switches, the same configuration cannot be used for port and interface template on the switch.

Information About Interface Templates

About Interface Templates

An interface template is a container of configurations or policies that can be applied to specific ports. When an interface template is applied to an access port, it impacts all traffic that is exchanged on the port.

There are two types of interface templates; user and builtin templates. Builtin templates are created by the system.

You can modify builtin templates. If you delete a modified builtin template the system restores the original definition of the template.

The following are the available builtin templates:

- AP_INTERFACE_TEMPLATE (Access Point)
- DMP_INTERFACE_TEMPLATE (Digital Media Player)
- IP_CAMERA_INTERFACE_TEMPLATE
- IP_PHONE_INTERFACE_TEMPLATE
- LAP_INTERFACE_TEMPLATE (Lightweight Access Point)
- MSP_CAMERA_INTERFACE_TEMPLATE
- MSP_VC_INTERFACE_TEMPLATE (Video Conferencing)
- PRINTER_INTERFACE_TEMPLATE
- ROUTER_INTERFACE_TEMPLATE
- SWITCH_INTERFACE_TEMPLATE
- TP_INTERFACE_TEMPLATE (TelePresence)

Following is an example of a builtin interface template:

```
Template Name       : IP_CAMERA_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 spanning-tree portfast
 spanning-tree bpduguard enable
 switchport mode access
 switchport block unicast
 switchport port-security
 mls qos trust dscp
 srr-queue bandwidth share 1 30 35 5
 priority-queue out
!
```

You can also create specific user templates with the commands that you want to include.

Note The template name must not contain spaces.

You can create an interface template using the **template** command in global configuration mode. In template configuration mode, enter the required commands. The following commands can be entered in template configuration mode:

Command	Description
access-session	Configures access session specific interface commands.
authentication	Configures authentication manager Interface Configuration commands.
carrier-delay	Configures delay for interface transitions.
dampening	Enables event dampening.
default	Sets a command to its defaults.
description	Configures interface-specific description.
dot1x	Configures interface configuration commands for IEEE 802.1X.
hold-queue	Sets hold queue depth.
ip	Configures IP template.
keepalive	Enables keepalive.
load-interval	Specifies interval for load calculation for an interface.
mab	Configures MAC authentication bypass Interface.
mls	Enables multilayer switching configurations. This command is available on the following devices in template configuration mode: • Cisco Catalyst 2960-S Series Switches • Cisco Catalyst 2960-X Series Switches • Cisco Industrial Ethernet 3000 Series Switches
peer	Configures peer parameters for point to point interfaces.
priority-queue	To set the priority-queue size for a template. This command is available on the following devices in template configuration mode: • Cisco Catalyst 2960-S Series Switches • Cisco Catalyst 2960-X Series Switches • Cisco Industrial Ethernet 3000 Series Switches

Command	Description
queue-set	Configures the QoS queue set on a template. This command is available on the following devices in template configuration mode: • Cisco Catalyst 2960-S Series Switches • Cisco Catalyst 2960-X Series Switches • Cisco Industrial Ethernet 3000 Series Switches
radius-server	Enables RADIUS server configurations. This command is available on the following devices in template configuration mode: • Catalyst 4500E Supervisor Engine 7-E • Catalyst 4500E Supervisor Engine 7L-E • Catalyst 4500E Supervisor Engine 8-E • Catalyst 4500-X Series Switches
service-policy	Configures CPL service policy.
source	Gets configurations from another source.
spanning-tree	Configures spanning tree subsystem
storm-control	Configures storm control.
subscriber	Configures subscriber inactivity timeout value.
switchport	Sets switching mode configurations
trust	Sets trust value for the interface.

Note System builtin templates are not displayed in the running configuration. These templates show up in the running configuration only if you edit them.

Binding an Interface Template to a Target

Each template can be bound to a target. Template binding or sourcing can be either static or dynamic. Static binding of a template involves binding the template to a target, like an interface. Only one template can be bound at a time using static binding. Static binding of another template to the same target will unbind the previously bound template. To configure static binding, use the **source template** command in interface configuration mode.

Any number of templates can be bound dynamically to a target. To configure dynamic binding using builtin policy maps and parameter maps, enable the autoconf feature using the **autoconf enable** command.

Note You can have statically and dynamically bind templates on the same interface at a time.

Priority for Configurations Using Interface Templates

Configuration applied through dynamically-bound templates has the highest priority, followed by configuration applied directly on the interface, and then configuration applied through statically-bound templates. When similar commands are present at different priority levels, the one at the highest priority is applied. If a configuration at a higher priority level is not applied, then the configuration with the next highest priority is applied to the target.

Multiple templates can be dynamically bound to a target. When multiple templates are dynamically bound, the template that is applied last has the highest priority.

To delete a template, you must remove the binding to all targets. If you bind a template that does not exist, a new template is created with no configurations.

How to Configure Interface Templates

Configuring Interface Templates

Perform the following task to create user interface templates:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **template** *name*
4. **load-interval** *interval*
5. **description** *description*
6. **keepalive** *number*
7. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**template** *name* **Example:** `Device(config)# template user-template1`	Creates a user template and enters template configuration mode. **Note** Builtin template are system-generated.
Step 4	**load-interval** *interval* **Example:** `Device(config-template)# load-interval 60`	Configures the sampling interval for statistics collections on the template. **Note** Builtin template are system-generated.
Step 5	**description** *description* **Example:** `Device(config-template)# description This is a user template`	Configures the description for the template.
Step 6	**keepalive** *number* **Example:** `Device(config-template)# Keepalive 60`	Configures the keepalive timer.
Step 7	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Static Binding for Interface Templates

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **source template** *name*
5. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/12`	Specifies the interface type and number and enters interface configuration mode.
Step 4	**source template** *name* **Example:** `Device(config-if)# source template user-template1`	Statically applies an interface template to a target.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Example

To verify static binding use the **show running-config interface** *int-name* and the **show derived-config interface** *int-name* commands.

```
Device# show running-config interface GigabitEthernet 1/0/12

Building configuration...

Current configuration : 71 bytes
!
interface GigabitEthernet1/0/12
source template user-template1
end

Device# show derived-config interface GigabitEthernet 1/0/12
Building configuration...

Derived configuration : 108 bytes
!
interface GigabitEthernet1/0/12
description This is a user template
load-interval 60
keepalive 60
end
```

Configuring Dynamic Binding of Interface Templates

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service-policy type control subscriber** *policymap-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 4/0/1`	Specifies the interface type and number and enters interface configuration mode.
Step 4	**service-policy type control subscriber** *policymap-name* **Example:** `Device(config-if)# service-policy type control subscriber POLICY-Gi1/0/12`	Dynamically applies an interface template to a target.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying an Interface Template

Use one or more of the commands listed below to verify the interface template configuration.

SUMMARY STEPS

1. **enable**

2. **show template interface all** {**all** | **binding** {*temp-name* | **all** | **target** *int-name*} | **brief** }
3. **show template interface source** {**built-in** [**original**] | **user**}{*temp-name* | **all**}}
4. **show template service**{**all** | **binding target** *int-name* | **brief** | **source** {**aaa** | **built-in** | **user** {**temp-name** | **all**}}

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show template interface all** {**all** | **binding** {*temp-name* | **all** | **target** *int-name*} | **brief** }
Shows all interface template configurations.

Step 3 **show template interface source** {**built-in** [**original**] | **user**}{*temp-name* | **all**}}
Shows interface template source configurations.

Step 4 **show template service**{**all** | **binding target** *int-name* | **brief** | **source** {**aaa** | **built-in** | **user** {**temp-name** | **all**}}
Shows all interface template service configurations.

Verifying Interface User Templates

Verifying all Builtin Templates

Verifying all Builtin Templates on Cisco Catalyst 2960-S Series Switches , Cisco Catalyst 2960-X Series Switches, Cisco Industrial Ethernet 3000 Series Switches

Verifying all Interface Templates Binding for all templates

Verifying Static Template Binding for a Target Interface

Verifying Dynamic Template Binding for all templates

Verifying Template Binding for a Target Interface

```
Device# show template interface source user all
    Template Name : TEST-1
    Template Definition:
    load-interval 60
    description TEST_1_TEMPLATE
    keepalive 200
    !
    Template Name : TEST-2
```

```
    Template Definition:
    load-interval 60
    description TEST-1_TEMPLATE
    keepalive 200

Device#  show template interface source built-in all

Building configuration...

Template Name : AP_INTERFACE_TEMPLATE
Modified : No
Template Definition :
switchport mode trunk
switchport nonegotiate
service-policy input AutoConf-4.0-Trust-Cos-Input-Policy
service-policy output AutoConf-4.0-Output-Policy
!
Template Name : DMP_INTERFACE_TEMPLATE
Modified : No
Template Definition :
switchport mode access
switchport block unicast
switchport port-security
spanning-tree portfast
spanning-tree bpduguard enable
service-policy input AutoConf-4.0-Trust-Dscp-Input-Policy
service-policy output AutoConf-4.0-Output-Policy
!
Template Name : IP_CAMERA_INTERFACE_TEMPLATE
Modified : No
Template Definition :
switchport mode access
switchport block unicast
switchport port-security
spanning-tree portfast
spanning-tree bpduguard enable
service-policy input AutoConf-4.0-Trust-Dscp-Input-Policy
service-policy output AutoConf-4.0-Output-Policy
!
Template Name : IP_PHONE_INTERFACE_TEMPLATE
Modified : No
Template Definition :
switchport mode access
switchport block unicast
switchport port-security maximum 3
switchport port-security maximum 2 vlan access
switchport port-security violation restrict
switchport port-security aging time 2
switchport port-security aging type inactivity
switchport port-security
storm-control broadcast level pps 1k
storm-control multicast level pps 2k
storm-control action trap
spanning-tree portfast
spanning-tree bpduguard enable
service-policy input AutoConf-4.0-CiscoPhone-Input-Policy
service-policy output AutoConf-4.0-Output-Policy
ip dhcp snooping limit rate 15
load-interval 30
!
Template Name : LAP_INTERFACE_TEMPLATE
Modified : No
Template Definition :
switchport mode access
switchport block unicast
```

```
switchport port-security violation protect
switchport port-security aging time 2
switchport port-security aging type inactivity
switchport port-security
storm-control broadcast level pps 1k
storm-control multicast level pps 2k
storm-control action trap
spanning-tree portfast
spanning-tree bpduguard enable
ip dhcp snooping limit rate 15
load-interval 30
!
Template Name : MSP_CAMERA_INTERFACE_TEMPLATE
Modified : No
Template Definition :
switchport mode access
switchport block unicast
switchport port-security
spanning-tree portfast
spanning-tree bpduguard enable
!
Template Name : MSP_VC_INTERFACE_TEMPLATE
Modified : No
Template Definition :
switchport mode access
switchport port-security
spanning-tree portfast
spanning-tree bpduguard enable
load-interval 30
!
Template Name : PRINTER_INTERFACE_TEMPLATE
Modified : No
Template Definition :
switchport mode access
switchport port-security maximum 2
switchport port-security
spanning-tree portfast
spanning-tree bpduguard enable
load-interval 60
!
Template Name : ROUTER_INTERFACE_TEMPLATE
Modified : No
Template Definition :
switchport mode trunk
spanning-tree portfast trunk
spanning-tree bpduguard enable
service-policy input AutoConf-4.0-Trust-Cos-Input-Policy
service-policy output AutoConf-4.0-Output-Policy
!
Template Name : SWITCH_INTERFACE_TEMPLATE
Modified : No
Template Definition :
switchport mode trunk
service-policy input AutoConf-4.0-Trust-Cos-Input-Policy
service-policy output AutoConf-4.0-Output-Policy
!
Template Name : TP_INTERFACE_TEMPLATE
Modified : No
Template Definition :
switchport mode access
switchport port-security maximum 3
switchport port-security maximum 2 vlan access
switchport port-security violation restrict
switchport port-security aging time 2
```

```
switchport port-security aging type inactivity
switchport port-security
storm-control broadcast level pps 1k
storm-control multicast level pps 2k
storm-control action trap
spanning-tree portfast
spanning-tree bpduguard enable
service-policy input AutoConf-4.0-Trust-Dscp-Input-Policy
service-policy output AutoConf-4.0-Output-Policy
ip dhcp snooping limit rate 15
load-interval 30
!
end
```

```
Device# show template interface source built-in all

Building configuration...

Template Name       : AP_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 switchport mode trunk
 switchport nonegotiate
 mls qos trust cos
 srr-queue bandwidth share 1 30 35 5
 priority-queue out
!
Template Name       : DMP_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 spanning-tree portfast
 spanning-tree bpduguard enable
 switchport mode access
 switchport block unicast
 switchport port-security
 mls qos trust dscp
 srr-queue bandwidth share 1 30 35 5
 priority-queue out
!
Template Name       : IP_CAMERA_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 spanning-tree portfast
 spanning-tree bpduguard enable
 switchport mode access
 switchport block unicast
 switchport port-security
 mls qos trust dscp
 srr-queue bandwidth share 1 30 35 5
 priority-queue out
!
Template Name       : IP_PHONE_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 spanning-tree portfast
 spanning-tree bpduguard enable
 switchport mode access
 switchport block unicast
 switchport port-security maximum 3
 switchport port-security maximum 2 vlan access
 switchport port-security violation  restrict
 switchport port-security aging time 2
 switchport port-security aging type inactivity
 switchport port-security
 storm-control broadcast level pps 1k
```

```
 storm-control multicast level pps 2k
 storm-control action trap
 mls qos trust cos
 service-policy input AUTOCONF-SRND4-CISCOPHONE-POLICY
 ip dhcp snooping limit rate 15
 load-interval 30
 srr-queue bandwidth share 1 30 35 5
 priority-queue out
!
Template Name       : LAP_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 spanning-tree portfast
 spanning-tree bpduguard enable
 switchport mode access
 switchport block unicast
 switchport port-security violation  protect
 switchport port-security aging time 2
 switchport port-security aging type inactivity
 switchport port-security
 storm-control broadcast level pps 1k
 storm-control multicast level pps 2k
 storm-control action trap
 mls qos trust dscp
 ip dhcp snooping limit rate 15
 load-interval 30
 srr-queue bandwidth share 10 10 60 20
 priority-queue out
!
Template Name       : MSP_CAMERA_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 spanning-tree portfast
 spanning-tree bpduguard enable
 switchport mode access
 switchport block unicast
 switchport port-security
!
Template Name       : MSP_VC_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 spanning-tree portfast
 spanning-tree bpduguard enable
 switchport mode access
 switchport block unicast
 switchport port-security violation  restrict
 switchport port-security aging time 2
 switchport port-security aging type inactivity
 switchport port-security
 ip dhcp snooping limit rate 15
 load-interval 30
!
Template Name       : PRINTER_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 spanning-tree portfast
 spanning-tree bpduguard enable
 switchport mode access
 switchport port-security maximum 2
 switchport port-security
 load-interval 60
!
Template Name       : ROUTER_INTERFACE_TEMPLATE
Modified            : No
```

```
Template Definition :
 spanning-tree portfast trunk
 spanning-tree bpduguard enable
 switchport mode trunk
 mls qos trust dscp
 srr-queue bandwidth share 1 30 35 5
 priority-queue out
!
Template Name       : SWITCH_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 switchport mode trunk
 mls qos trust cos
 srr-queue bandwidth share 1 30 35 5
 priority-queue out
!
Template Name       : TP_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 spanning-tree portfast
 spanning-tree bpduguard enable
 switchport mode access
 switchport port-security maximum 3
 switchport port-security maximum 2 vlan access
 switchport port-security violation  restrict
 switchport port-security aging time 2
 switchport port-security aging type inactivity
 switchport port-security
 storm-control broadcast level pps 1k
 storm-control multicast level pps 2k
 storm-control action trap
 ip dhcp snooping limit rate 15
 load-interval 30
!
End
```

```
Device# show template interface binding all
    Template-name               Source      Method          Interface
    =============               ====      ==========      ==========
IP_PHONE_INTERFACE_TEMPLATE    Built-in    Dynamic        Gi1/0/1, Gi1/0/2, Gi1/0/3
                                                          Gi1/0/4, Gi1/0/5, Gi1/0/6
                                                          Gi1/0/7, Gi1/0/8, Gi1/0/9
                                                          Gi1/0/10, Gi1/0/11, Gi1/0/12
                                                          Gi1/0/13, Gi1/0/14, Gi1/0/15
                                                          Gi1/0/16, Gi1/0/17, Gi1/0/18
                                                          Gi1/0/19, Gi1/0/20, Gi1/0/21
                                                          Gi1/0/22, Gi1/0/23, Gi1/0/24
                                                          Gi1/1/1, Gi1/1/2, Gi1/1/3

IP_PHONE_INTERFACE_TEMPLATE    Built-in     Static        Gi4/0/4
```

```
Device# show template interface binding target GigabitEthernet 1/0/4
    Interface            Method       Source            Template
    =========           ==========    =====             =========
    Gi1/0/4              Dynamic      built-in          IP_PHONE_INTERFACE_TEMPLATE
                         Static       user              TEST
                         Dynamic      Modified-built-in TEST
```

```
Device# show template service all

    User-defined template:
    ======================

    Template Name      : SVC-1
    Template Definition:
```

```
    vlan 100
    access-group acl1

    built-in template:
    ======================

    Template Name      : SVC-2
    Template Definition:
    vlan 100
    access-group acl1

    aaa downloaded template:
    ========================
    Template Name      : SVC-2
    Template Definition:
    vlan 100
    access-group acl1
```

```
Device# show template binding target GigabitEthernet 1/0/4

  Interface Templates:
    Interface            method       Source             Template
    =========           ==========    =====              =========
    Gi1/0/4              Dynamic      built-in           IP_PHONE_INTERFACE_TEMPLATE
                         Static       user               TEST
                         Dynamic      Modified-built-in TEST
  Service Templates:
   Template            Source         Session-Mac
   ========            ====           ================
   SVC1                user           aa-bb-cc-dd-ee-ff
   SVC2                built-in       ab-ab-ab-ab-ab-ab
   SVC3                aaa            ac-ac-ac-ac-ac-ac
```

Configuration Examples for Interface Templates

Example: Configuring User Interface Templates

Example: Configuring User Templates

```
Device# enable
Device (config)# configure terminal
Device(config)# template user-template1
Device(config-template)# load-interval 60
Device(config-template)# description This is a user template
Device(config-template)# Keepalive 60
Device(config)# end
```

Example:Sourcing Interface Templates

```
Device> enable
Device# configure terminal
Device(config)# interface fastethernet 4/0/0
Device(config-if)# source template user-template1
Device(config-if)# end
```

Example: Dynamically Binding Interface Templates

```
Device> enable
Device# configure terminal
Device(config)# interface GigabitEthernet 4/0/1
Device(config-if)# service-policy type control subscriber POLICY_Gi1/0/12
Device(config-if)# end
```

Additional References for Interface Templates

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Identity-Based Networking Services commands	Cisco IOS Identity-Based Networking Services Command Reference
Autoconf	"Autoconf" module in `Identity-Based Networking Services Configuration Guide`.

Technical Assistance

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

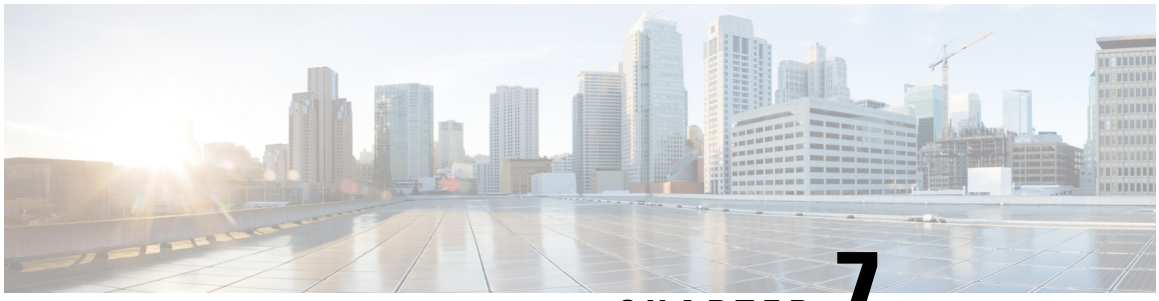
Feature Information for Interface Templates

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

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

Table 7: Feature Information for Interface Templates

Feature Name	Releases	Feature Information
Interface Templates	Cisco IOS XE 3.6E	An interface template provides a mechanism to configure multiple commands at the same time and associate it with a target such as an interface. In Cisco IOS XE Release 3.6E, this feature is supported on the following platforms: • Cisco Catalyst 4500E Supervisor Engine 6-E • Cisco Catalyst 4500E Supervisor Engine 6L-E • Cisco Catalyst 4500E Supervisor Engine 7-E • Cisco Catalyst 4500E Supervisor Engine 7L-E • Cisco Catalyst 4500E Supervisor Engine 8-E • Cisco Catalyst 4500-X Series Switches • Cisco Catalyst 3850 Series Switches • Cisco Catalyst 3750E Series Switches • Cisco Catalyst 3750-X Series Switches • Cisco Catalyst 3650 Series Switches • Cisco Catalyst 3650C Series Switches The following commands were introduced or modified: **access-session** , **authentication**, **carrier-delay**, **dampening**, **default**, **description**, **dot1x**, **hold-queue**, **ip** , **keepalive**, **load-interval**, **mab**, **mls**, **peer**, **priority-queue**, **queue-set**, **radius-server**, **service-policy type control subscriber**, **source**, **spanning-tree**, **storm-control**, **subscriber**, **switchport**, **trust**.



CHAPTER 7

Autoconf

Autoconf is a solution that can be used to manage port configurations for data or voice VLAN, quality of service (QoS) parameters, storm control, and MAC-based port security on end devices that are deployed in the access layer of a network.

Finding Feature Information

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

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

Prerequisites for Autoconf

- Before enabling Autoconf, disable the Auto SmartPort (ASP) macro, device classifier, and then access the session monitor.

Restrictions for Autoconf

- Interface templates are not applicable for wireless sessions.
- When the Autoconf feature is enabled using the **autoconf enable** command, the default Autoconf service policy is applied to all interfaces. No other service policy can be applied globally using the **service-policy**

command. To apply a different service policy, you must disable Autoconf on that interface. When a service policy is applied globally, you must disable it before enabling the Autoconf feature.

- When both local (interface-level) and global service policies exist, the local policy take precedence. Events in the local service policy are handled and the global service policy is not applied. The global service policy comes into effect only when the local policy is removed.
- Service templates cannot be applied to interfaces, and interface templates cannot be applied to service instances.
- Only one service template can be nested inside an interface template.

Information About Autoconf

Benefits of Autoconf

The Autoconf feature permits hardbinding between the end device and the interface. Autoconf falls under the umbrella of the Smart Operations solution. Smart Operations is a comprehensive set of capabilities that can simplify and improve LAN switch deployment. Smart Operations help organizations deliver operational excellence and scale services on the network.

The Autoconf feature automatically applies the needed configurations on the device ports to enable the efficient performance of each directly connected end device using a set of interface configurations that are configured inside an interface template.

- Autoconf efficiently applies commands to an interface because the parser does not need to parse each command each time.
- Configurations that are applied through the Autoconf feature can be reliably removed from a port without impacting previous or subsequent configurations on the port.
- The Autoconf feature provides built-in and user-defined configurations using interface and service templates. Configurations applied through templates can be centrally updated with a single operation.
- Using the Autoconf feature, a configuration can be applied to ports and access sessions.
- The Autoconf feature reduces ongoing maintenance for devices and attached end devices by making them intuitive and autoconfigurable. This reduces operation expenses (OPEX) and lowers the total cost of ownership (TCO).

Identity Session Management and Templates

A key advantage of the Autoconf feature is that the core session management capability is decoupled from the application-specific logic; thus, allowing the same framework to be used regardless of the criteria for policy determination or the nature of the policies applied.

The identity session management infrastructure allows configurations and/or policies to be applied as templates.

Both service and interface templates are named containers of configuration and policy. Service templates may be applied only to access sessions, while interface templates may be applied only to ports. When a service template is applied to an access session, the contained configuration/policy is applied only to the target session and has no impact on other sessions that may be hosted on the same access port. Similarly, when an interface template is applied to an access port, it impacts all traffic exchanged on the port.

The Autoconf feature uses a set of built-in maps and built-in templates. The built-in templates are designed based on best practices for interface configurations. Built-in templates can be modified by the user to include customized configurations, limiting the need to create a new template.

The templates created by users are referred to as user-defined templates. User-defined templates can be defined on the device and can be mapped to any built-in or user-defined trigger.

Use the **show derived-config** command, to view the overall applied configurations applied by Autoconf template and manual configuration. The interface commands shown in the output of **show running-config interface** *type number* command are not necessarily the operational configuration. The Autoconf feature dynamically applies a template to the interface, and overrides any conflicting static configuration that is already applied.

Autoconf Operation

Autoconf uses the Device Classifier to identify the end devices that are connected to a port.

The Autoconf feature uses the device classification information gleaned from Cisco Discovery Protocol, LLDP, DHCP, MAC addresses, and the Organizationally Unique Identifier (OUI) that is identified by the Device Classifier.

The Device Classifier provides improved device classification capabilities and accuracy, and increased device visibility for enhanced configuration management.

Device classification is enabled when you enable the Autoconf feature using **autoconf enable** command in global configuration mode .

The device detection acts as an event trigger, which in turn applies the appropriate automatic template to the interface.

The Autoconf feature is based on a three-tier hierarchy.

- A policy map identifies the trigger type for applying the Autoconf feature.
- A parameter map identifies the appropriate template that must be applied, based on the end device.
- The templates contain the configurations to be applied.

The Autoconf built-in templates and triggers perform the these three steps automatically.

The Autoconf feature provides the following built-in templates:

- AP_INTERFACE_TEMPLATE
- DMP_INTERFACE_TEMPLATE
- IP_CAMERA_INTERFACE_TEMPLATE
- IP_PHONE_INTERFACE_TEMPLATE
- LAP_INTERFACE_TEMPLATE
- MSP_CAMERA_INTERFACE_TEMPLATE
- MSP_VC_INTERFACE_TEMPLATE
- PRINTER_INTERFACE_TEMPLATE
- ROUTER_INTERFACE_TEMPLATE

- SWITCH_INTERFACE_TEMPLATE
- TP_INTERFACE_TEMPLATE

Note By default built-in templates are not displayed under running configuration. The built-in templates show in the running configuration only if you edit them.

The template that is selected is based on parameter map information applied to an interface. This information can be based on the following criteria:

- End Device type
- MAC address
- OUI
- User role
- Username

The Autoconf feature provides one built-in parameter map BUILTIN_DEVICE_TO_TEMPLATE with the following configuration:

```
Parameter-map name: BUILTIN_DEVICE_TO_TEMPLATE
 Map: 10 map device-type regex "Cisco-IP-Phone"
  Action(s):
   20 interface-template IP_PHONE_INTERFACE_TEMPLATE
 Map: 20 map device-type regex "Cisco-IP-Camera"
  Action(s):
   20 interface-template IP_CAMERA_INTERFACE_TEMPLATE
 Map: 30 map device-type regex "Cisco-DMP"
  Action(s):
   20 interface-template DMP_INTERFACE_TEMPLATE
 Map: 40 map oui eq "00.0f.44"
  Action(s):
   20 interface-template DMP_INTERFACE_TEMPLATE
 Map: 50 map oui eq "00.23.ac"
  Action(s):
   20 interface-template DMP_INTERFACE_TEMPLATE
 Map: 60 map device-type regex "Cisco-AIR-AP"
  Action(s):
   20 interface-template AP_INTERFACE_TEMPLATE
 Map: 70 map device-type regex "Cisco-AIR-LAP"
  Action(s):
   20 interface-template LAP_INTERFACE_TEMPLATE
 Map: 80 map device-type regex "Cisco-TelePresence"
  Action(s):
   20 interface-template TP_INTERFACE_TEMPLATE
 Map: 90 map device-type regex "Surveillance-Camera"
  Action(s):
   10 interface-template MSP_CAMERA_INTERFACE_TEMPLATE
 Map: 100 map device-type regex "Video-Conference"
  Action(s):
   10 interface-template MSP_VC_INTERFACE_TEMPLATE
```

Note Use the **show parameter-map type subscriber attribute-to-service All** command to view the configuration for the built-in parameter map.

The Autoconf feature provides one built-in policy map BUILTIN_AUTOCONF_POLICY with the following configuration:

```
BUILTIN_AUTOCONF_POLICY
  event identity-update match-all
    10 class always do-until-failure
      10 map attribute-to-service table BUILTIN_DEVICE_TO_TEMPLATE
```

Note Use the **show policy-map type control subscriber BUILTIN_AUTOCONF_POLICY** command to view the configuration for the built-in policy map.

You can also manually create policy maps, parameter maps, and templates.

When a trigger is created that is based on specific user information, a local 802.1X Cisco Identity Services Engine (ISE) server authenticates it ensuring the security of the operation.

An interface template can be dynamically activated (on an interface) using any of the following methods:

- RADIUS CoA—While Change of Authorization (CoA) commands are targeted to one or more access sessions, any referenced template must be applied to the interface hosting the referenced session.
- RADIUS Access-Accept for client authentication or authorization—Any referenced interface template returned in an Access-Accept must be applied to the port that is hosting the authorized access session.
- Service template—If an interface template is referenced in a service template that is either locally defined or sourced from the AAA server, the interface template must be applied to the interface hosting any access-session on which the service template is applied (add a new command for interface template reference from within a locally defined service template).
- Subscriber control-policy action—A mapping action under the subscriber control policy activates service and/or interface template (as referenced in a parameter map) based on the type of filter, and removes any templates associated with a previous policy.
- Device-to-template parameter map—A subscriber parameter map that allows the filter type to service and/or interface template mappings to be specified in an efficient and readable manner.

Advantages of Using Templates

Using templates for autoconfiguration has the following benefits:

- Templates are parsed once when they are being defined. This makes dynamic application of the templates very efficient.
- Templates can be applied to an Ethernet interface that is connected to an end device, based on the type of the end device.
- Service templates allow the activation of session-oriented features, whereas interface templates apply configurations to the interface that is hosting a session.
- Service templates are applied to access sessions and hence only impact the traffic exchanged with a single endpoint on a port.
- Startup and running configurations of the device are not modified by the dynamic application of the template.

- Policy application is synchronized with the access-session life cycle, which is tracked by the framework by using all available techniques, including just link-up/link-down.
- Templates can be updated with a single operation. All applied instances of the templates are updated.
- Constituent commands of the templates do not appear in the running configuration.
- Templates can be removed with no impact on previous or subsequent configurations.
- Template application is acknowledged, allowing for synchronization and performing remedial actions where failures occur.
- Data VLAN, quality of service (QoS) parameters, storm control, and MAC-based port security are configured automatically based on the end device that is connected to the switch.
- The switch port is cleaned up completely by removing configurations when the device is disconnected from a port.
- Human error is reduced in the installation and configuration process.

Autoconf Functionality

The Autoconf feature is disabled by default in global configuration mode. When you enable the Autoconf i feature in global configuration mode, it is enabled by default at the interface level. The built-in template configurations are applied based on the end devices detected on all interfaces.

Use the **access-session inherit disable autoconf** command to manually disable Autoconf at the interface level, even when Autoconf is enabled at the global level.

If you disable Autoconf at the global level, all interface-level configurations are disabled.

Global	Interface Level	AutoConf Status
Disable	Disable	No automatic configurations are applied when an end device is connected.
Enable	Enabled by default	If Autoconf is enabled at the global level, it is enabled at the interface level by default. Built-in template configurations are applied based on the end devices that are detected on all interfaces.
Enable	Disable	Enabled at global level. Disabled at interface level. No automatic configurations are applied when an end device is connected to the interface on which Autoconf is disabled.

Autoconf allows you to retain the template even when the link to the end device is down or the end device is disconnected, by configuring the Autoconf sticky feature. Use the **access-session interface-template sticky** command to configure the Autoconf sticky feature in global configuration mode. The Autoconf sticky feature avoids the need for detecting the end device and applying the template every time the link flaps or device is removed and connected back.

The **access-session interface-template sticky** command is mandatory to apply an inbuilt template that contains **access-session** commands on an interface. Configure the **access-session interface-template sticky** command to apply interface template on a port using a service policy.

If you want to disable the Autoconf feature on a specific interface, use the **access-session inherit disable interface-template-sticky** command in interface configuration mode.

How to Configure Autoconf

Applying a Built-in Template to an End Device

The following task shows how to apply a built-in template on an interface that is connected to an end device, for example, a Cisco IP phone.

Before you begin

Make sure that the end device, for example, a Cisco IP phone, is connected to the switch port.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **autoconf enable**
4. **end**
5. (Optional) **show device classifier attached interface** *interface-type interface-number*
6. **show template binding target** *interface-type interface-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device(config)# configure terminal`	Enters global configuration mode.
Step 3	**autoconf enable** **Example:** `Device(config)# autoconf enable`	Enables the Autoconf feature.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.
Step 5	(Optional) **show device classifier attached interface** *interface-type interface-number* **Example:** `Device# show device classifier attached interface Gi3/0/26`	Displays whether the end device is classified by the device classifier with correct attributes.

	Command or Action	Purpose
Step 6	**show template binding target** *interface-type interface-number* **Example:** `Device# show template binding target gi3/0/26`	Displays the configuration applied through the template on the interface.

Verifying th device classification of an End Device

Verifying the Interface Template on an Interface

Verifying the Interface Configuration

Verifying Interface Configuration for Cisco IOS 4500 Series, Cisco IOS 3650 Series, Cisco IOS 3560 Series, and Cisco IOS 2960 Switches

Verifying Global Configuration after Applying Autoconf

The following example shows that an IP phone is classified by the Device Classifier with correct attributes:

```
Device# show device classifier attached interface GigabitEthernet 3/0/26

Summary:

MAC_Address       Port_Id       Profile Name                        Device Name
==============    ==========    ==================                  ==================
0026.0bd9.7bbb    Gi3/0/26      Cisco-IP-Phone-7962                 Cisco IP Phone 7962
```

The following example shows that a built-in interface template is applied on the interface:

```
Device# show template binding target GigabitEthernet 3/0/26

Interface Templates
==================
Interface: Gi4/0/11
Method              Source            Template-Name
------              ------            -------------
dynamic             Built-in          IP_PHONE_INTERFACE_TEMPLATE
```

The following example shows how to verify the interface configuration after the interface template is applied to the IP phone connected to the GigabitEthernet interface 3/0/26 :

```
Device# show running-config interface GigabitEthernet 3/0/26
Building configuration...

Current configuration : 624 bytes
!
interface GigabitEthernet3/0/26
!
End

Device# show derived-config interface GigabitEthernet 3/0/26

Building configuration...

Derived configuration : 649 bytes
```

```
!
interface GigabitEthernet3/0/26
 switchport mode access
 switchport block unicast
 switchport port-security maximum 3
 switchport port-security maximum 2 vlan access
 switchport port-security violation restrict
 switchport port-security aging time 2
 switchport port-security aging type inactivity
 switchport port-security
 load-interval 30
 storm-control broadcast level pps 1k
 storm-control multicast level pps 2k
 storm-control action trap
 spanning-tree portfast
 spanning-tree bpduguard enable
 service-policy input AutoConf-4.0-CiscoPhone-Input-Policy
 service-policy output AutoConf-4.0-Output-Policy
 ip dhcp snooping limit rate 15
end
```

The following example shows how to verify the interface configuration:

```
Device# show template interface source built-in all
Building configuration...

Template Name       : AP_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 switchport mode trunk
 switchport nonegotiate
 mls qos trust cos
 srr-queue bandwidth share 1 30 35 5
 priority-queue out
!
Template Name       : DMP_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 spanning-tree portfast
 spanning-tree bpduguard enable
 switchport mode access
 switchport block unicast
 switchport port-security
 mls qos trust dscp
 srr-queue bandwidth share 1 30 35 5
 priority-queue out
!
Template Name       : IP_CAMERA_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 spanning-tree portfast
 spanning-tree bpduguard enable
 switchport mode access
 switchport block unicast
 switchport port-security
 mls qos trust dscp
 srr-queue bandwidth share 1 30 35 5
 priority-queue out
!
Template Name       : IP_PHONE_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 spanning-tree portfast
 spanning-tree bpduguard enable
 switchport mode access
```

```
 switchport block unicast
 switchport port-security maximum 3
 switchport port-security maximum 2 vlan access
 switchport port-security violation  restrict
 switchport port-security aging time 2
 switchport port-security aging type inactivity
 switchport port-security
 storm-control broadcast level pps 1k
 storm-control multicast level pps 2k
 storm-control action trap
 mls qos trust cos
 service-policy input AUTOCONF-SRND4-CISCOPHONE-POLICY
 ip dhcp snooping limit rate 15
 load-interval 30
 srr-queue bandwidth share 1 30 35 5
 priority-queue out
!
Template Name       : LAP_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 spanning-tree portfast
 spanning-tree bpduguard enable
 switchport mode access
 switchport block unicast
 switchport port-security violation  protect
 switchport port-security aging time 2
 switchport port-security aging type inactivity
 switchport port-security
 storm-control broadcast level pps 1k
 storm-control multicast level pps 2k
 storm-control action trap
 mls qos trust dscp
 ip dhcp snooping limit rate 15
 load-interval 30
 srr-queue bandwidth share 10 10 60 20
 priority-queue out
!
Template Name       : MSP_CAMERA_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 spanning-tree portfast
 spanning-tree bpduguard enable
 switchport mode access
 switchport block unicast
 switchport port-security
!
Template Name       : MSP_VC_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 spanning-tree portfast
 spanning-tree bpduguard enable
 switchport mode access
 switchport block unicast
 switchport port-security violation  restrict
 switchport port-security aging time 2
 switchport port-security aging type inactivity
 switchport port-security
 ip dhcp snooping limit rate 15
 load-interval 30
!
Template Name       : PRINTER_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 spanning-tree portfast
```

```
 spanning-tree bpduguard enable
 switchport mode access
 switchport port-security maximum 2
 switchport port-security
 load-interval 60
!
Template Name       : ROUTER_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 spanning-tree portfast trunk
 spanning-tree bpduguard enable
 switchport mode trunk
 mls qos trust dscp
 srr-queue bandwidth share 1 30 35 5
 priority-queue out
!
Template Name       : SWITCH_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 switchport mode trunk
 mls qos trust cos
 srr-queue bandwidth share 1 30 35 5
 priority-queue out
!
Template Name       : TP_INTERFACE_TEMPLATE
Modified            : No
Template Definition :
 spanning-tree portfast
 spanning-tree bpduguard enable
 switchport mode access
 switchport port-security maximum 3
 switchport port-security maximum 2 vlan access
 switchport port-security violation  restrict
 switchport port-security aging time 2
 switchport port-security aging type inactivity
 switchport port-security
 storm-control broadcast level pps 1k
 storm-control multicast level pps 2k
 storm-control action trap
 ip dhcp snooping limit rate 15
 load-interval 30
!

Device# show running config
class-map match-any AutoConf-4.0-Scavenger-Queue
 match dscp cs1
 match cos  1
 match access-group name AutoConf-4.0-ACL-Scavenger
class-map match-any AutoConf-4.0-VoIP
 match dscp ef
 match cos  5
class-map match-any AutoConf-4.0-Control-Mgmt-Queue
 match cos  3
 match dscp cs7
 match dscp cs6
 match dscp cs3
 match dscp cs2
 match access-group name AutoConf-4.0-ACL-Signaling
class-map match-any AutoConf-4.0-Multimedia-Conf
 match dscp af41
 match dscp af42
 match dscp af43
class-map match-all AutoConf-4.0-Broadcast-Vid
 match dscp cs5
class-map match-any AutoConf-4.0-Bulk-Data
```

```
 match dscp af11
 match dscp af12
 match dscp af13
class-map match-all AutoConf-4.0-Realtime-Interact
 match dscp cs4
class-map match-any AutoConf-4.0-VoIP-Signal
 match dscp cs3
 match cos  3
class-map match-any AutoConf-4.0-Trans-Data-Queue
 match cos  2
 match dscp af21
 match dscp af22
 match dscp af23
 match access-group name AutoConf-4.0-ACL-Transactional-Data
class-map match-any AutoConf-4.0-VoIP-Data
 match dscp ef
 match cos  5
class-map match-any AutoConf-4.0-Multimedia-Stream
 match dscp af31
 match dscp af32
 match dscp af33
class-map match-all AutoConf-4.0-Internetwork-Ctrl
 match dscp cs6
class-map match-all AutoConf-4.0-VoIP-Signal-Cos
 match cos  3
class-map match-any AutoConf-4.0-Multimedia-Stream-Queue
 match dscp af31
 match dscp af32
 match dscp af33
class-map match-all AutoConf-4.0-Network-Mgmt
 match dscp cs2
class-map match-all AutoConf-4.0-VoIP-Data-Cos
 match cos  5
class-map match-any AutoConf-4.0-Priority-Queue
 match cos  5
 match dscp ef
 match dscp cs5
 match dscp cs4
class-map match-any AutoConf-4.0-Bulk-Data-Queue
 match cos  1
 match dscp af11
 match dscp af12
 match dscp af13
 match access-group name AutoConf-4.0-ACL-Bulk-Data
class-map match-any AutoConf-4.0-Transaction-Data
 match dscp af21
 match dscp af22
 match dscp af23
class-map match-any AutoConf-4.0-Multimedia-Conf-Queue
 match cos  4
 match dscp af41
 match dscp af42
 match dscp af43
 match access-group name AutoConf-4.0-ACL-Multimedia-Conf
class-map match-all AutoConf-4.0-Network-Ctrl
 match dscp cs7
class-map match-all AutoConf-4.0-Scavenger
 match dscp cs1
class-map match-any AutoConf-4.0-Signaling
 match dscp cs3
 match cos  3
!
!
policy-map AutoConf-4.0-Cisco-Phone-Input-Policy
```

```
 class AutoConf-4.0-VoIP-Data-Cos
  set dscp ef
  police cir 128000 bc 8000
   exceed-action set-dscp-transmit cs1
   exceed-action set-cos-transmit 1
 class AutoConf-4.0-VoIP-Signal-Cos
  set dscp cs3
  police cir 32000 bc 8000
   exceed-action set-dscp-transmit cs1
   exceed-action set-cos-transmit 1
 class class-default
  set dscp default
  set cos 0
policy-map AutoConf-4.0-Output-Policy
 class AutoConf-4.0-Scavenger-Queue
  bandwidth remaining percent 1
 class AutoConf-4.0-Priority-Queue
  priority
  police cir percent 30 bc 33 ms
 class AutoConf-4.0-Control-Mgmt-Queue
  bandwidth remaining percent 10
 class AutoConf-4.0-Multimedia-Conf-Queue
  bandwidth remaining percent 10
 class AutoConf-4.0-Multimedia-Stream-Queue
  bandwidth remaining percent 10
 class AutoConf-4.0-Trans-Data-Queue
  bandwidth remaining percent 10
    dbl
 class AutoConf-4.0-Bulk-Data-Queue
  bandwidth remaining percent 4
    dbl
 class class-default
  bandwidth remaining percent 25
    dbl
policy-map AutoConf-DMP
 class class-default
  set dscp cs2
policy-map AutoConf-IPVSC
 class class-default
  set cos dscp table AutoConf-DscpToCos
policy-map AutoConf-4.0-Input-Policy
 class AutoConf-4.0-VoIP
 class AutoConf-4.0-Broadcast-Vid
 class AutoConf-4.0-Realtime-Interact
 class AutoConf-4.0-Network-Ctrl
 class AutoConf-4.0-Internetwork-Ctrl
 class AutoConf-4.0-Signaling
 class AutoConf-4.0-Network-Mgmt
 class AutoConf-4.0-Multimedia-Conf
 class AutoConf-4.0-Multimedia-Stream
 class AutoConf-4.0-Transaction-Data
 class AutoConf-4.0-Bulk-Data
 class AutoConf-4.0-Scavenger
```

Applying a Modified Built-in Template to an End Device

The following task shows how to modify a built-in template when multiple wireless access points and IP cameras are connected to a switch.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **template** *template-name*
4. **switchport access vlan** *vlan-id*
5. **description** *description*
6. **exit**
7. **autoconf enable**
8. **end**
9. **show template interface binding all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device(config)# configure terminal`	Enters global configuration mode.
Step 3	**template** *template-name* **Example:** `Device(config)# template AP_INTERFACE_TEMPLATE`	Enters template configuration mode for the builtin template.
Step 4	**switchport access vlan** *vlan-id* **Example:** `Device(config-template)# switchport access vlan 20`	Sets the VLAN when the interface is in access mode.
Step 5	**description** *description* **Example:** `Device(config-template)# description modifiedAP`	Modifies the description of the built-in template.
Step 6	**exit** **Example:** `Device(config-template)# exit`	Exits template configuration mode and enters global configuration mode.
Step 7	**autoconf enable** **Example:** `Device(config)# autoconf enable`	Enables the Autoconf feature.
Step 8	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
Step 9	**show template interface binding all** **Example:** `Device# show template interface binding all`	Displays whether the template is applied on the interface.

Verifying the Device classification of an End Device

Verifying the Interface Template on an Interface

The following example shows that the IP camera and access points are classified by the Device Classifier with correct attributes:

```
Device# show device classifier attached detail

DC default profile file version supported = 1

Detail:
MAC_Address     Port_Id    Cert Parent Proto    ProfileType     Profile Name
Device_Name
=============================================================================================================
001d.a1ef.23a8  Gi1/0/7     30    3    C     M    Default      Cisco-AIR-AP-1130      cisco
 AIR-AP1131AG-A-K9
001e.7a26.eb05  Gi1/0/30    70    2    C     M    Default      Cisco-IP-Camera        Cisco
 IP Camera
```

The following example shows that a built-in interface template is applied on the interface:

```
Device# show template interface binding all

Template-Name                   Source            Method          Interface
-------------                   ------            ------          ---------
IP_CAMERA_INTERFACE_TEMPLATE     Built-in          dynamic         Gi1/0/30
AP_INTERFACE_TEMPLATE          Modified-Built-in   dynamic         Gi1/0/7
```

Migrating from ASP to Autoconf

Before you begin

Verify that the AutoSmart Port (ASP) macro is running using the **show running-config | include macro auto global** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no macro auto global processing**
4. **exit**
5. **clear macro auto configuration all**
6. **configure terminal**

7. **autoconf enable**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no macro auto global processing** **Example:** `Device(config)# no macro auto global processing`	Disables ASP on a global level.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**clear macro auto configuration all** **Example:** `Device# clear macro auto configuration all`	Clears macro configurations for all interfaces.
Step 6	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 7	**autoconf enable** **Example:** `Device(config)# autoconf enable`	Enables the Autoconf feature.
Step 8	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for Autoconf

Example: Applying a Built-in Template to an End Device

The following example shows how to apply a built-in template to an end device connected to an interface.

```
Device> enable
Device(config)# configure terminal
Device(config)# autoconf enable
Device(config)# end
Device# show device classifier attached interface Gi3/0/26
Device# show template binding target GigabitEthernet 3/0/26
```

Example: Applying a Modified Built-in Template to an End Device

The following example shows how to modified built-in template and verify the configuration:

```
Device> enable
Device(config)# configure terminal
Device(config)# template AP_INTERFACE_TEMPLATE
Device(config-template)# switchport access vlan 20
Device(config-template)# description modifiedAP
Device(config-template)# exit
Device(config)# autoconf enable
Device(config)# end
Device# show template interface binding all
```

Example: Migrating from ASP Macros to Autoconf

The following example shows how to migrate from ASP to Autoconf:

```
Device> enable
Device# configure terminal
Device(config)# no macro auto global processing
Device(config)# exit
Device# clear macro auto configuration all
Device# configure terminal
Device(config)# autoconf enable
Device(config)# end
```

Additional References for Autoconf

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
Cisco identity-based networking services commands	Cisco IOS Identity-Based Networking Services Command Reference
Interface Templates	"Interface Templates" module in `Identity-Based Networking Services Configuration Guide`.

Standards and RFCs

Standard/RFC	Title
IEEE 802.1X	*Port Based Network Access Control*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

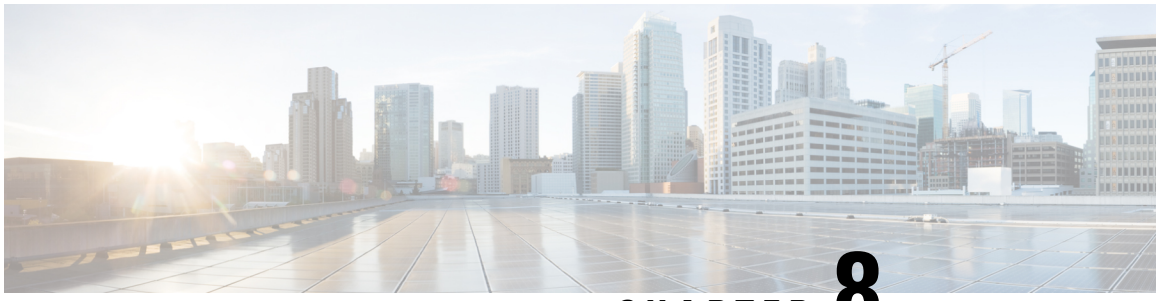
Feature Information for Autoconf

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

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

Table 8: Feature Information for Autoconf

Feature Name	Releases	Feature Information
Autoconf	Cisco IOS XE 3.6E	Autoconf is a solution that can be used to manage port configurations for data or voice VLANs, quality of QoS parameters, storm control, and MAC-based port security on end devices that are deployed in the access layer of a network. The Autoconf feature automatically applies the configurations needed on the device ports to enable the efficient performance of each directly connected end device using a set of interface configurations that are configured inside an interface template. This mechanism ensures that no configurations are needed from the end device. In Cisco IOS XE Release 3.6E, this feature is supported on the following platforms: • Cisco Catalyst 4500E Supervisor Engine 6-E • Cisco Catalyst 4500E Supervisor Engine 6L-E • Cisco Catalyst 4500E Supervisor Engine 7-E • Cisco Catalyst 4500E Supervisor Engine 7L-E • Cisco Catalyst 4500E Supervisor Engine 8-E • Cisco Catalyst 4500-X Series Switches • Cisco Catalyst 3850 Series Switches • Cisco Catalyst 3750E Series Switches • Cisco Catalyst 3750-X Series Switches • Cisco Catalyst 3650 Series Switches • Cisco Catalyst 3650C Series Switches The following commands were added or modified: **autoconf enable**, **map attribute-to-service (autoconf)**, **map device-type (service-template)**, **parameter-map type subscriber (service-template)**, **show parameter-map type subscriber attribute-to-service all**, **show template interface**.



CHAPTER 8

eEdge Integration with MACsec

The Media Access Control Security (MACsec) standard is the IEEE 802.1AE standard for authenticating and encrypting packets between two MACsec-capable devices. The eEdge Integration with MACsec feature allows you to integrate the MACsec standard with enterprise edge (eEdge) devices to enhance Session Aware Networking capabilities. Session Aware Networking provides a policy and identity-based framework for edge devices to deliver flexible and scalable services to subscribers.

Finding Feature Information

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

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

Prerequisites for eEdge Integration with MACsec

- Layer 2 encryption protocols like the IEEE 802.1AE Media Access Control Security (MACsec) standard must register with the eEdge session manager to receive disconnect notifications and perform cleanup.
- You must provision one virtual interface per secure association.

Restrictions for eEdge Integration with MACsec

- The Media Access Control Security (MACsec) standard is supported only in single-host and multihost modes. If a link layer security policy is configured as must-secure and the host mode is not configured as a single host or a multihost, the connection is closed.
- The MACsec standard is not supported in multi-authentication mode.
- The MACsec standard supports the 802.1AE encryption with MACsec Key Agreement (MKA) only on downlink ports for encryption between a MACsec-capable device and host devices.

Information About eEdge Integration with MACsec

Overview of MACsec

Media Access Control Security (MACsec) is the IEEE 802.1AE standard for authenticating and encrypting packets between two MACsec-capable devices. Implementing the MACsec encryption standard enables support for the 802.1AE encryption with MACsec Key Agreement (MKA) on downlink ports for encryption between a MACsec-capable device and host devices. The MACsec-capable device also supports MACsec link layer device-to-device security by using Cisco TrustSec Network Device Admission Control (NDAC) and the Security Association Protocol (SAP) key exchange. Link layer security includes both packet authentication between devices and MACsec encryption between devices (encryption is optional).

MACsec Standard Encryption

The Media Access Control Security (MACsec) standard provides data link layer encryption over wired networks by using out-of-band methods for encryption keying. The MACsec Key Agreement (MKA) protocol provides the required session keys and manages the encryption keys. MKA and MACsec are implemented after a successful authentication by using the 802.1X Extensible Authentication Protocol (EAP) framework. Only host-facing links (links between network access devices and endpoint devices such as a PC or an IP phone) can be secured using MACsec.

A device that uses MACsec accepts either MACsec or non-MACsec frames, depending on the policy associated with the client. MACsec frames are encrypted and protected with an integrity check value (ICV). When the device receives frames from the client, it decrypts them and calculates the correct ICV by using session keys provided by MKA. The device compares the calculated value of the ICV to the ICV within the frame. If they are not identical, the frame is dropped. The device also encrypts and adds an ICV to any frame that is sent over a secured port (the access point used to provide the secure MAC service to a client) using the current session key.

The MKA protocol manages the encryption keys used by the underlying MACsec protocol. The basic requirements of MKA are defined in 802.1X-2010. The MKA protocol extends 802.1X to allow peer discovery with confirmation of mutual authentication and sharing of MACsec secret keys to protect data exchanged by peers.

EAP Implementation of MKA

The Extensible Authentication Protocol (EAP) framework implements MKA as a newly defined EAP-over-LAN (EAPOL) packet. EAP authentication produces a master session key (MSK) that is shared by both partners

in the data exchange. Entering the EAP session ID generates a secure connectivity association key name (CKN). Because the device is the authenticator, it is also the key server, generating a random 128-bit secure association key (SAK), which it sends it to the client partner. The client is never a key server and can only interact with a single MKA entity, the key server. After key derivation and generation, the device sends periodic transports to the partner at a default interval of 2 seconds.

The packet body in an EAPOL Protocol Data Unit (PDU) is referred to as a MACsec Key Agreement PDU (MKPDU). MKA sessions and participants are deleted when the MKA lifetime (6 seconds) passes and no MKPDU is received from a participant. For example, if a client disconnects, the participant on the device continues to operate MKA until 6 seconds have elapsed after the last MKPDU is received from the client.

Integrating eEdge with MACsec

In enterprise edge (eEdge) devices the encryption protocol is implemented as a session manager client and an Enterprise Policy Manager (EPM) plugin.

When you implement EPM plugin and the session manager client, the data link layer security is implemented as an EPM feature, which returns an asynchronous result to EPM when authentication is successful.

When the data link layer security user profile is applied and sessions are configured as either must-secure or should-secure using the **linksec policy** {**must-secure** | **should-secure**} command, the MACsec Key Agreement (MKA) processing starts.

The Media Access Control Security (MACsec) encryption standard is a data link layer security protocol. On eEdge devices, you must explicitly configure the protocol within a service template and an associated policy action.

The eEdge Integration with MACsec feature enables integrating the MACsec standard on a device using a service template.

How to Configure eEdge Integration with MACsec

Integrating eEdge with MACsec

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **service-template** *template-name*
4. **linksec policy** {**must-not-secure** | **must-secure** | **should-secure**}
5. **exit**
6. **policy-map type control subscriber** *control-policy-name*
7. **event authentication-success** [**match-all** | **match-any**]
8. *priority-number* **class** {*control-class-name* | **always**} [**do-all** | **do-until-failure** | **do-until-success**]
9. *action-number* **activate** {**policy type control subscriber** *control-policy-name* | **service-template** *template-name* [**aaa-list** *list-name*] [**precedence** [**replace-all**]}
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**service-template** *template-name* **Example:** `Device(config)# service-template dot1x-macsec-policy`	Defines a template that contains a set of service policy attributes to apply to subscriber sessions and enters service template configuration mode.
Step 4	**linksec policy** {**must-not-secure** | **must-secure** | **should-secure**} **Example:** `Device(config-service-template)# linksec policy must-secure`	Sets the link security policy as must-secure. • Must-secure policy authorizes the eEdge device port only if a secure MACsec session is established.
Step 5	**exit** **Example:** `Device(config-service-template)# exit`	Exits service template configuration mode and returns to global configuration mode.
Step 6	**policy-map type control subscriber** *control-policy-name* **Example:** `Device(config)# policy-map type control subscriber cisco-subscriber`	Defines a control policy for subscriber sessions and enters control policy-map event configuration mode.
Step 7	**event authentication-success** [**match-all** | **match-any**] **Example:** `Device(config-event-control-policymap)# event authentication-success match-all`	Specifies the type of event that triggers actions in a control policy if all authentication events are a match and enters control policy-map class configuration mode.
Step 8	*priority-number* **class** {*control-class-name* | **always**} [**do-all** | **do-until-failure** | **do-until-success**] **Example:** `Device(config-class-control-policymap)# 10 class always do-until-failure`	Specifies that the control class should execute the actions in a control policy, in the specified order, until one of the actions fails, and enters control policy-map action configuration mode.
Step 9	*action-number* **activate** {**policy type control subscriber** *control-policy-name* | **service-template** *template-name* [**aaa-list** *list-name*] [**precedence** [**replace-all**]} **Example:** `Device(config-action-control-policymap)# 10 activate service-template dot1x-macsec-policy`	Activates a control policy on a subscriber session.

	Command or Action	Purpose
Step 10	**end** **Example:** `Device(config-action-control-policymap)# end`	Exits control policy-map action configuration mode and enters privileged EXEC mode.

Identifying Link Layer Security Failures

SUMMARY STEPS

1. **configure terminal**
2. **class-map type control subscriber** {**match-all** | **match-any** | **match-none**} *control-class-name*
3. **match authorization-failure** {**domain-change-failed** | **linksec-failed**}
4. **exit**
5. **policy-map type control subscriber** *control-policy-name*
6. **event authentication-failure** [**match-all** | **match-any**]
7. *priority-number* **class** {*control-class-name* | **always**} [**do-all** | **do-until-failure** | **do-until-success**]
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**class-map type control subscriber** {**match-all** | **match-any** | **match-none**} *control-class-name* **Example:** `Device(config)# class-map type control subscriber match-all linksec-failed`	Creates a control class, which defines the conditions under which the actions of a control policy are executed and enters control class-map filter configuration mode.
Step 3	**match authorization-failure** {**domain-change-failed** | **linksec-failed**} **Example:** `Device(config-filter-control-classmap)# match authorization-failure linksec-failed`	Configures a match condition in a control class based on the type of authorization failure received from an authorization failed event of a link layer security failure.
Step 4	**exit** **Example:** `Device(config-class-control-policymap)# exit`	Exits control class-map filter configuration mode and enters global configuration mode.
Step 5	**policy-map type control subscriber** *control-policy-name* **Example:**	Defines a control policy for subscriber sessions and enters control policy-map event configuration mode.

	Command or Action	Purpose
	`Device(config)# policy-map type control subscriber cisco-subscriber`	
Step 6	**event authentication-failure** [**match-all** | **match-any**] **Example:** `Device(config-event-control-policymap)# event authentication-failure match-all`	Specifies the type of event that triggers actions in a control policy if session authentication fails and enters control policy-map class configuration mode.
Step 7	*priority-number* **class** {*control-class-name* | **always**} [**do-all** | **do-until-failure** | **do-until-success**] **Example:** `Device(config-class-control-policymap)# 10 class linksec-failed do-until-failure`	Specifies that the control class must execute the actions in a control policy, in the specified order, until one of the actions fails and enters control policy-map action configuration mode.
Step 8	**end** **Example:** `Device(config-action-control-policymap)# end`	Exits control policy-map action configuration mode and enters privileged EXEC mode.

Configuration Examples for eEdge Integration with MACsec

Example: Integrating eEdge with MACsec

```
Device> enable
Device# configure terminal
Device(config)# service-template dot1x-macsec-policy
Device(config-service-template)# linksec policy must-secure
Device(config-service-template)# exit
Device(config)# policy-map type control subscriber cisco-subscriber
Device(config-event-control-policymap)# event authentication-success match-all
Device(config-class-control-policymap)# 10 class always do-until-failure
Device(config-action-control-policymap)# 10 activate service-template dot1x-macsec-policy
Device(config-action-control-policymap)# end
```

Example: Identifying Linksec Failures

```
Device# configure terminal
Device(config)# class-map type control subscriber match-all linksec-failure
Device(config-filter-control-classmap)# match authorization-failure linksec-failed
Device(config-class-control-classmap)# exit
Device(config)# policy-map type control subscriber cisco-subscriber
Device(config-event-control-policymap)# event authentication-failure match-all
Device(config-class-control-policymap)# 10 class linksec-failed do-until-failure
Device(config-action-control-policymap)# end
```

Additional References for eEdge Integration with MACsec

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Identity-Based Networking Services Command Reference	Cisco IOS Identity-Based Networking Services Command Reference

Standards and RFCs

Standard/RFC	Title
IEEE 802.1AE Standard	802.1AE - Media Access Control (MAC) Security

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

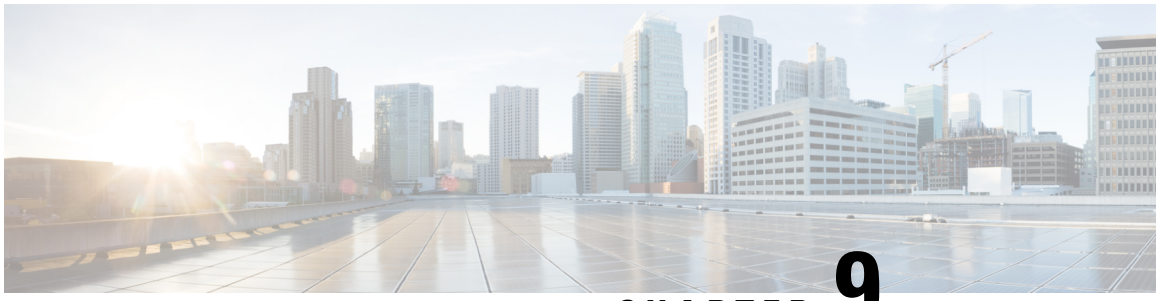
Feature Information for eEdge Integration with MACsec

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

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

Table 9: Feature Information for eEdge Integration with MACsec

Feature Name	Releases	Feature Information
eEdge Integration with MACsec	Cisco IOS XE Release 3.5E Cisco IOS XE Release 3.6E	The eEdge Integration with MACsec feature allows you to integrate the MACsec standard with enterprise edge (eEdge) devices to enhance Session Aware Networking capabilities. In Cisco IOS XE 3.5E, this feature is supported on the following platforms: • Cisco Catalyst 4500-X Series Switches • Cisco Catalyst 4500E Supervisor Engine 7L-E In Cisco IOS XE Release 3.6E, this feature is supported on Cisco Catalyst 3850 Series Switches The following commands were introduced or modified: **linksec policy**, **match authorization-failure**.



CHAPTER 9

Critical Voice VLAN Support

The Critical Voice VLAN Support feature directs phone traffic to the configured voice VLAN of a port if the authentication server becomes unreachable.

With normal network connectivity, when an IP phone successfully authenticates on a port, the authentication server directs the phone traffic to the voice domain of the port. If the authentication server becomes unreachable, IP phones cannot authenticate the phone traffic. In multidomain authentication (MDA) mode or multiauthentication mode, you can configure the Critical Voice VLAN Support feature to direct phone traffic to the configured voice VLAN of the port. The phone is authorized as an unknown domain. Both data and voice are enabled for the phone.

Finding Feature Information

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

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

Restrictions for Critical Voice VLAN Support

- Different VLANs must be configured for voice and data.
- The voice VLAN must be configured on a device.
- The Critical Voice VLAN Support feature does not support standard Access Control Lists (ACLs) on the switch port.

Information About Critical Voice VLAN Support

Critical Voice VLAN Support in Multidomain Authentication Mode

If a critical voice VLAN is deployed using an interface in multidomain authentication (MDA) mode, the host mode is changed to multihost and the first phone device is installed as a static forwarding entries. Any additional phone devices are installed as dynamic forwarding entry in the Host Access Table (HAT).

For further information about host modes, see the *802.1X Authentication Services Configuration Guide*.

Note If a critical port is already authorized and reauthentication occurs, the switch puts the port in the critical-authentication state in the current VLAN, which might be the one previously assigned by the RADIUS server.

Note Inaccessible authentication bypass is compatible with guest VLAN. When a guest VLAN is enabled on a 802.1X port, the features interact as follows: if all RADIUS servers are not available and if a client is connected to a critical port and was previously assigned to a guest VLAN, the switch keeps the port in the guest VLAN.

Critical Voice VLAN Support in Multiauthentication Mode

If the critical authentication feature is deployed in multiauthentication mode, only one phone device will be allowed and a second phone trying to authorize will trigger a violation.

The **show authentication sessions** command displays the critical voice client data. A critically authorized voice client in multiauthentication host mode will be in the “authz success” and “authz fail” state.

Note If critical voice is required, then critical data should be configured too. Otherwise, the critical voice client will be displayed in the “authz fail” state while the voice VLAN will be open.

Critical Voice VLAN Support in a Service Template

On enterprise Edge (eEdge) devices, the critical access of phones is configured by activating a critical service template when the authentication server becomes unreachable. The voice feature plug-in registers with the Enterprise Policy Manager (EPM) by using an authentication, authorization, and accounting (AAA) voice attribute, and it allows unconditional access to the voice VLAN while the AAA services are unavailable.

To enable critical voice VLAN support, the critical authentication of phones must be configured using a combination of control policy rules and a service template.

When the authentication server is unavailable and the host is unauthorized, the AAA attribute device-traffic-type is not populated. The phone is authorized as an unknown domain, and both the data and voice VLAN are enabled for this device, allowing the device to handle voice traffic.

How to Configure Critical Voice VLAN Support

Configuring a Voice VLAN in a Service Template

Perform this task on a port to configure critical voice VLAN support using a service template.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **service-template** *template-name*
4. **vlan** *vlan-id*
5. **exit**
6. **service-template** *template-name*
7. **voice vlan**
8. **exit**
9. **class-map type control subscriber** {**match-all** | **match-any** | **match-none**} *control-class-name*
10. **match result-type** [**method** {**dot1x** | **mab** | **webauth**}] *result-type*
11. **match authorization-status** {**authorized** | **unauthorized**}
12. **exit**
13. **class-map type control subscriber** {**match-all** | **match-any** | **match-none**} *control-class-name*
14. **match result-type** [**method** {**dot1x** | **mab** | **webauth**}] *result-type*
15. **match authorization-status** {**authorized** | **unauthorized**}
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**service-template** *template-name* **Example:** `Device(config)# service-template SERVICE-TEMPLATE`	Defines a template that contains a set of service policy attributes to apply to subscriber sessions and enters service template configuration mode.
Step 4	**vlan** *vlan-id* **Example:**	Assigns a VLAN to a subscriber session.

	Command or Action	Purpose
	`Device(config-service-template)# vlan 116`	
Step 5	**exit** **Example:** `Device(config-service-template)# exit`	Exits service template configuration mode and returns to global configuration mode.
Step 6	**service-template** *template-name* **Example:** `Device(config)# service-template CRITICAL-VOICE`	Defines a template that contains a set of service policy attributes to apply to subscriber sessions and enters service template configuration mode.
Step 7	**voice vlan** **Example:** `Device(config-service-template)# voice vlan`	Assigns a critical voice VLAN to a subscriber session.
Step 8	**exit** **Example:** `Device(config-service-template)# exit`	Exits service template configuration mode and returns to global configuration mode.
Step 9	**class-map type control subscriber** {**match-all** | **match-any** | **match-none**} *control-class-name* **Example:** `Device(config)# class-map type control subscriber match-all AAA-SVR-DOWN-UNAUTHD-HOST`	Creates a control class, which defines the conditions under which the actions of a control policy are executed and enters control class-map filter configuration mode.
Step 10	**match result-type** [**method** {**dot1x** | **mab** | **webauth**}] *result-type* **Example:** `Device(config-filter-control-classmap)# match result-type aaa-timeout`	Creates a condition that returns true based on the specified authentication result.
Step 11	**match authorization-status** {**authorized** | **unauthorized**} **Example:** `Device(config-filter-control-classmap)# match authorization-status unauthorized`	Creates a condition that returns true based on the authorization status of a session.
Step 12	**exit** **Example:** `Device(config-filter-control-classmap)# exit`	Exits control class-map filter configuration mode and returns to global configuration mode.
Step 13	**class-map type control subscriber** {**match-all** | **match-any** | **match-none**} *control-class-name* **Example:** `Device(config)# class-map type control subscriber match-all AAA-SVR-DOWN-AUTHD-HOST`	Creates a control class, which defines the conditions under which the actions of a control policy are executed and enters control class-map filter configuration mode.

	Command or Action	Purpose
Step 14	**match result-type** [**method** {**dot1x** | **mab** | **webauth**}] *result-type* **Example:** `Device(config-filter-control-classmap)# match result-type aaa-timeout`	Creates a condition that returns true based on the specified authentication result.
Step 15	**match authorization-status** {**authorized** | **unauthorized**} **Example:** `Device(config-filter-control-classmap)# match authorization-status authorized`	Creates a condition that returns true based on the authorization status of a session.
Step 16	**end** **Example:** `Device(config-filter-control-classmap)# end`	Exits control class-map filter configuration mode and returns to privileged EXEC mode.

Activating Critical Voice VLAN

Perform the following task to activate a critical voice VLAN that is configured on a service template.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type control subscriber** *control-policy-name*
4. **event authentication-failure** [**match-all** | **match-first**]
5. *priority-number* **class** {*control-class-name* | **always**} [**do-all** | **do-until-failure** | **do-until-success**]
6. *action-number* **activate** {**policy type control subscriber** *control-policy-name* | **service-template** *template-name* [**aaa-list** *list-name*] [**precedence** [**replace-all**]]}
7. *action-number* **activate** {**policy type control subscriber** *control-policy-name* | **service-template** *template-name* [**aaa-list** *list-name*] [**precedence** [**replace-all**]]}
8. *action-number* **authorize**
9. *action-number* **pause reauthentication**
10. **exit**
11. *priority-number* **class** {*control-class-name* | **always**} [**do-all** | **do-until-failure** | **do-until-success**]
12. *action-number* **pause reauthentication**
13. **end**

DETAILED STEPS

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

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map type control subscriber** *control-policy-name* **Example:** `Device(config)# policy-map type control subscriber cisco-subscriber`	Defines a control policy for subscriber sessions and enters control policy-map event configuration mode.
Step 4	**event authentication-failure** [**match-all** | **match-first**] **Example:** `Device(config-event-control-policymap)# event authentication-failure match-first`	Specifies the type of event that triggers actions in a control policy if all authentication events are a match and enters control policy-map class configuration mode.
Step 5	*priority-number* **class** {*control-class-name* | **always**} [**do-all** | **do-until-failure** | **do-until-success**] **Example:** `Device(config-class-control-policymap)# 10 class AAA-SVR-DOWN-UNAUTHD-HOST do-until-failure`	Specifies that the control class should execute the actions in a control policy, in the specified order, until one of the actions fails, and enters control policy-map action configuration mode.
Step 6	*action-number* **activate** {**policy type control subscriber** *control-policy-name* | **service-template** *template-name* [**aaa-list** *list-name*] [**precedence** [**replace-all**]]} **Example:** `Device(config-action-control-policymap)# 10 activate service-template foo-DATA`	Activates a control policy associated with the VLAN on a subscriber session.
Step 7	*action-number* **activate** {**policy type control subscriber** *control-policy-name* | **service-template** *template-name* [**aaa-list** *list-name*] [**precedence** [**replace-all**]]} **Example:** `Device(config-action-control-policymap)# 10 activate service-template CRITICAL-VOICE`	Activates a control policy associated with the voice VLAN on a subscriber session.
Step 8	*action-number* **authorize** **Example:** `Device(config-action-control-policymap)# 30 authorize`	Initiates the authorization of a subscriber session.
Step 9	*action-number* **pause reauthentication** **Example:** `Device(config-action-control-policymap)# 40 pause reauthentication`	Pauses the reauthentication process after an authentication failure.

	Command or Action	Purpose
Step 10	**exit** **Example:** `Device(config-action-control-policymap)# exit`	Exits control policy-map action configuration mode and enters control policy-map class configuration mode.
Step 11	*priority-number* **class** {*control-class-name* | **always**} [**do-all** | **do-until-failure** | **do-until-success**] **Example:** `Device(config-class-control-policymap)# 20 class AAA-SVR-DOWN-AUTHD-HOST`	Specifies that the control class should execute the actions in a control policy, in the specified order, until one of the actions fails, and enters control policy-map action configuration mode.
Step 12	*action-number* **pause reauthentication** **Example:** `Device(config-action-control-policymap)# 10 pause reauthentication`	Pauses the reauthentication process after an authentication failure.
Step 13	**end** **Example:** `Device(config-action-control-policymap)# exit`	Exits control policy-map action configuration mode and enters privileged EXEC mode.

Configuration Examples for Critical Voice VLAN Support

Example: Configuring a Voice VLAN in a Service Template

```
Device> enable
Device# configure terminal
Device(config)# service-template SERVICE-TEMPLATE
Device(config-service-template)# vlan 116
Device(config-service-template)# exit
Device(config)# service-template CRITICAL-VOICE
Device(config-service-template)# voice vlan
Device(config-service-template)# exit
Device(config)# class-map type control subscriber match-all AAA-SVR-DOWN-UNAUTHD-HOST
Device(config-filter-control-classmap)# match result-type aaa-timeout
Device(config-filter-control-classmap)# match authorization-status unauthorized
Device(config-filter-control-classmap)# exit
Device(config)# class-map type control subscriber match-all AAA-SVR-DOWN-AUTHD-HOST
Device(config-filter-control-classmap)# match result-type aaa-timeout
Device(config-filter-control-classmap)# match authorization-status authorized
Device(config-filter-control-classmap)# end
```

Example: Activating a Critical Voice VLAN on a Service Template

```
Device> enable
Device# configure terminal
Device(config)# policy-map type control subscriber cisco-subscriber
Device(config-event-control-policymap)# event authentication-failure match-first
```

```
Device(config-class-control-policymap)# 10 class AAA-SVR-DOWN-UNAUTHD-HOST do-until-failure
Device(config-action-control-policymap)# 10 activate service-template SERVICE-TEMPLATE
Device(config-action-control-policymap)# 10 activate service-template CRITICAL-VOICE
Device(config-action-control-policymap)# 30 authorize
Device(config-action-control-policymap)# 40 pause reauthentication
Device(config-action-control-policymap)# exit
Device(config-class-control-policymap)# 20 class AAA-SVR-DOWN-AUTHD-HOST
Device(config-action-control-policymap)# 10 pause reauthentication
Device(config-action-control-policymap)# end
```

Additional References for Critical Voice VLAN Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco Identity-Based Networking Services commands	Cisco IOS Identity-Based Networking Services Command Reference

Standards and RFCs

Standard/RFC	Title
IEEE 802.1X	*Port based Network Access Control*

Technical Assistance

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

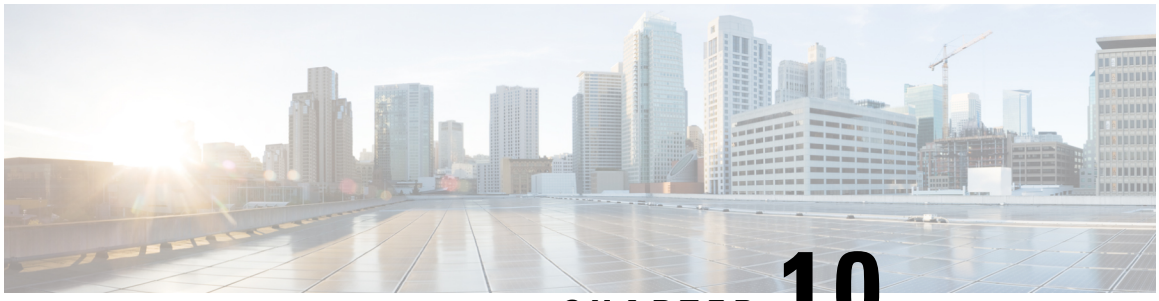
Feature Information for Critical Voice VLAN Support

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

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

Table 10: Feature Information for Critical Voice VLAN Support

Feature Name	Releases	Feature Information
Critical Voice VLAN Support	Cisco IOS XE Release 3.3SE	This feature enables critical voice VLAN support, which puts phone traffic into the configured voice VLAN of a port if the authentication server becomes unreachable. In Cisco IOS XE 3.3SE, this feature is supported on the following platforms: • Cisco 5700 Series Wireless LAN Controllers • Cisco Catalyst 3650 Series Switches • Cisco Catalyst 3850 Series Switches • Cisco Catalyst 4500E Supervisor Engine 7L-E The following command was added or modified: **voice vlan**



CHAPTER 10

Configuring Local Authentication Using LDAP

Local authentication using Lightweight Directory Access Protocol (LDAP) allows an endpoint to be authenticated using 802.1X, MAC authentication bypass (MAB), or web authentication with LDAP as a backend. Local authentication in Identity-Based Networking Services also supports associating an authentication, authorization, and accounting (AAA) attribute list with the local username. This module provides information about configuring local authentication for Identity-Based Networking Services.

Finding Feature Information

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

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

Information About Local Authentication Using LDAP

Local Authentication Using LDAP

Local authentication using LDAP allows an endpoint to be authenticated using 802.1X, MAB, or web authentication with LDAP as a backend. Local authentication also supports additional AAA attributes by associating an attribute list with a local username for wireless sessions.

AES Key Wrap

The Advanced Encryption Standard (AES) key wrap feature makes the shared secret between the controller and the RADIUS server more secure. AES key wrap is designed for Federal Information Processing Standards (FIPS) customers and requires a key-wrap compliant RADIUS authentication server.

How to Configure Local Authentication Using LDAP

Configuring Local Authentication Using LDAP

Perform this task to specify the AAA method list for local authentication and to associate an attribute list with a local username.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa local authentication** {*method-list-name* | **default**} **authorization** {*method-list-name* | **default**}
5. **username** *name* **aaa attribute list** *aaa-attribute-list* [**password** *password*]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables the authentication, authorization, and accounting (AAA) access control model.
Step 4	**aaa local authentication** {*method-list-name* | **default**} **authorization** {*method-list-name* | **default**} **Example:** `Device(config)# aaa local authentication default authorization default`	Specifies the method lists to use for local authentication and authorization from a LDAP server.
Step 5	**username** *name* **aaa attribute list** *aaa-attribute-list* [**password** *password*]	Associates a AAA attribute list with a local username.

	Command or Action	Purpose
	Example: `Device(config)# username USER_1 aaa attribute list LOCAL_LIST password CISCO`	
Step 6	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring MAC Filtering Support

Perform this task to set the RADIUS compatibility mode, the MAC delimiter, and the MAC address as the username to support MAC filtering.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa group server radius** *group-name*
5. **subscriber mac-filtering security-mode** {**mac** | **none** | **shared-secret**}
6. **mac-delimiter** {**colon** | **hyphen** | **none** | **single-hyphen**}
7. **exit**
8. **username** *mac-address* **mac** [**aaa attribute list** *aaa-attribute-list*]
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables the authentication, authorization, and accounting (AAA) access control model.
Step 4	**aaa group server radius** *group-name* **Example:** `Device(config)# aaa group server radius RAD_GROUP1`	Groups different RADIUS server hosts into distinct lists.

	Command or Action	Purpose
Step 5	**subscriber mac-filtering security-mode** {**mac** | **none** | **shared-secret**} **Example:** `Device(config-sg-radius)# subscriber mac-filtering security-mode mac`	Specifies the RADIUS compatibility mode for MAC filtering. • The default value is **none**.
Step 6	**mac-delimiter** {**colon** | **hyphen** | **none** | **single-hyphen**} **Example:** `Device(config-sg-radius)# mac-delimiter hyphen`	Specifies the MAC delimiter for RADIUS compatibility mode. • The default value is **none**.
Step 7	**exit** **Example:** `Device(config-sg-radius)# exit`	Exits server group configuration mode and returns to global configuration mode.
Step 8	**username** *mac-address* **mac** [**aaa attribute list** *aaa-attribute-list*] **Example:** `Device(config)# username 00-22-WP-EC-23-3C mac aaa attribute list AAA_list1`	Allows a MAC address to be used as the username for MAC filtering done locally.
Step 9	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Enabling AES Key Wrap

Advanced Encryption Standard (AES) key wrap makes the shared secret between the controller and the RADIUS server more secure. AES key wrap requires a key-wrap compliant RADIUS authentication server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **radius-server host** {*hostname* | *ip-address*} **key-wrap encryption-key** *encryption-key* **message-auth-code-key** *encryption-key* [**format** {**ascii** | **hex**}]
4. **aaa new-model**
5. **aaa group server radius** *group-name*
6. **server** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*]
7. **key-wrap enable**
8. **end**

DETAILED STEPS

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

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**radius-server host** {*hostname* | *ip-address*} **key-wrap encryption-key** *encryption-key* **message-auth-code-key** *encryption-key* [**format** {**ascii** | **hex**}] **Example:** `Device(config)# radius-server host 10.10.1.2 key-wrap encryption-key testkey99 message-auth-code-key testkey123`	Defines a RADIUS server host.
Step 4	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables the authentication, authorization, and accounting (AAA) access control model.
Step 5	**aaa group server radius** *group-name* **Example:** `Device(config)# aaa group server radius RAD_GROUP1`	Groups different RADIUS server hosts into distinct lists.
Step 6	**server** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*] **Example:** `Device(config-sg-radius)# server 10.10.1.2`	Specifies the IP address of the RADIUS server in the server group.
Step 7	**key-wrap enable** **Example:** `Device(config-sg-radius)# key-wrap enable`	Enables AES key wrap for this RADIUS server.
Step 8	**end** **Example:** `Device(config-sg-radius)# end`	Exits server group configuration mode and returns to privileged EXEC mode.

Configuration Examples for Local Authentication Using LDAP

Example: Configuring Local Authentication Using LDAP

The following example shows a configuration for local authentication:

```
!
username USER_1 password 0 CISCO
```

```
username USER_1 aaa attribute list LOCAL_LIST
aaa new-model
aaa local authentication EAP_LIST authorization EAP_LIST
!
```

Example: Configuring MAC Filtering Support

The following example shows a configuration for MAC filtering:

```
username 00-22-WP-EC-23-3C mac aaa attribute list AAA_list1
!
aaa new-model
aaa group server radius RAD_GROUP1
subscriber mac-filtering security-mode mac
mac-delimiter hyphen
```

Example: Configuring AES Key Wrap

The following example shows a configuration with key wrap enabled for a RADIUS server:

```
aaa new-model
aaa group server radius RAD_GROUP1
server 10.10.1.2
key-wrap enable
!
radius-server host 10.10.1.2
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Identity-Based Networking Services commands	Cisco IOS Identity-Based Networking Services Command Reference
Address Resolution Protocol (ARP) commands	Cisco IOS IP Addressing Services Command Reference
ARP configuration tasks	IP Addressing - ARP Configuration Guide
Authentication, authorization, and accounting (AAA) configuration tasks	Authentication Authorization and Accounting Configuration Guide
AAA commands	Cisco IOS Security Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 5176	Dynamic Authorization Extensions to RADIUS

Technical Assistance

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

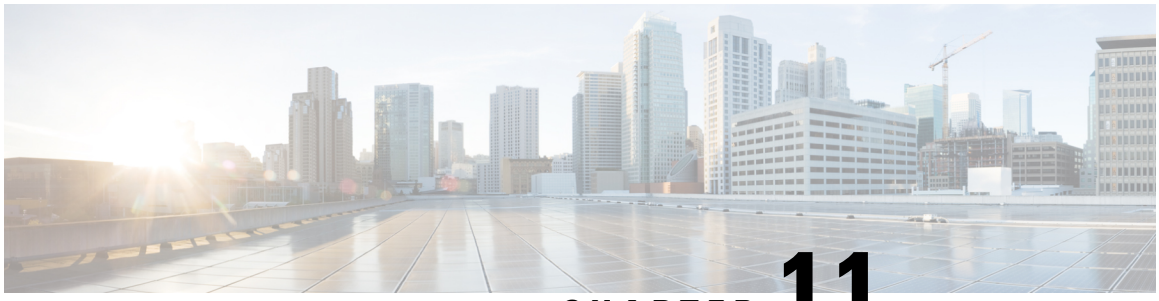
Feature Information for Local Authentication Using LDAP

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

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

Table 11: Feature Information for Local Authentication Using LDAP

Feature Name	Releases	Feature Information
Local Authentication Using LDAP	Cisco IOS XE Release 3.2SE	Introduces support for local authentication using Lightweight Directory Access Protocol (LDAP). In Cisco IOS XE 3.2SE, this feature is supported on the following platforms: • Cisco Catalyst 3650 Series Switches • Cisco Catalyst 3850 Series Switches • Cisco 5700 Wireless LAN Controllers The following commands were introduced or modified: **aaa local authentication**, **key-wrap enable**, **mac-delimiter**, **radius-server host**, **subscriber mac-filtering security-mode**, **username**.



CHAPTER 11

Wired Guest Access

The Wired Guest Access feature enables guest users of an enterprise network that supports both wired and wireless access to connect to the guest access network from a wired Ethernet connection. The wired Ethernet connection is designated and configured for guest access. Wired session guests on mobility agents are directed to a wireless guest controller in a demilitarized zone (DMZ) through a Control And Provisioning of Wireless Access Points (CAPWAP) tunnel.

Wired guest access can be configured in a dual-controller configuration that uses both an anchor controller and a foreign controller. A dual-controller configuration isolates wired guest access traffic.

Finding Feature Information

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

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

Restrictions for Wired Guest Access

- Tunneling of wired clients is not supported when the client is attached to a port at the Converged Access device that is configured for open mode.
- Tunneling of wired clients is not supported after successful web authentication at the Converged Access device because automated IP address reassignment is not supported after web-authentication.
- The Converged Access device supports network access only via the tunnel based on the web authentication that occurs at the controller.

- The Network Advertisement and Selection Protocol (NASP) is not supported for wired clients.
- High availability is not supported for wireless sessions. If the wireless controller fails while providing tunneled guest access for a wired client, the state is not automatically recovered.
- Inactivity aging is not enforced for a wired client that is provisioned to the wireless controller; for example, within a RADIUS Access-Accept request that is received after web authentication is performed at the controller.

Information About Wired Guest Access

Wired Guest Access Overview

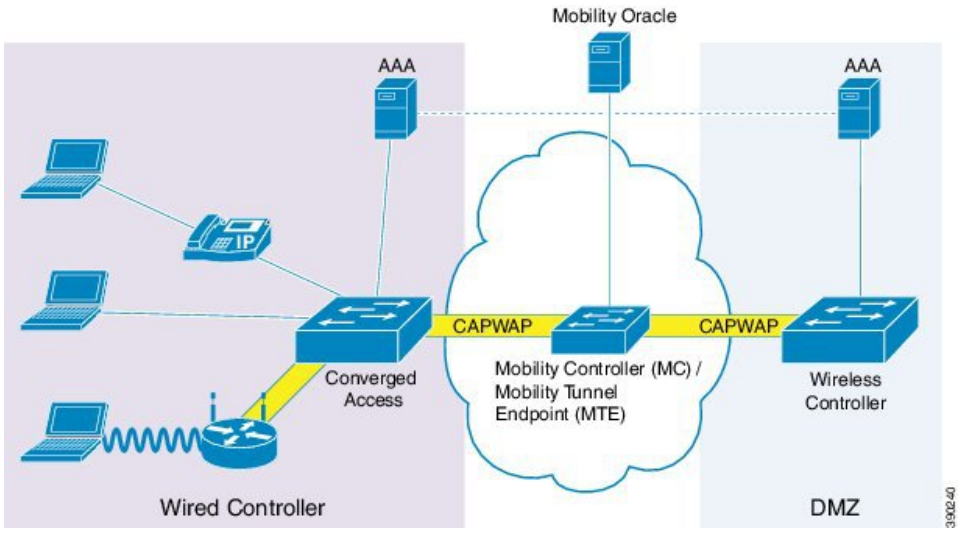
Enterprise networks that support both wired and wireless access need to provide guest services that are consistent across the two access media, from a perspective of both client experience and manageability. For wireless networks, guest traffic from a mobility anchor device is directed typically through a Control And Provisioning of Wireless Access Points (CAPWAP) tunnel to an array of controllers in the demilitarized zone (DMZ), where either web-authenticated access or open access is provided. Wired guest traffic can also be backhauled to the DMZ using more traditional tunneling mechanisms like Generic Routing Encapsulation (GRE). The Converged Access platforms, with converged wired and wireless access, can extend CAPWAP tunneling to wired guests also, allowing for very similar handling at the controller platform (in the DMZ) and reducing the provisioning overhead.

However, security remains an issue because it is not possible to determine, prior to authentication, whether a wired client is a guest or requires access to the corporate network. Consequently, the decision to tunnel a wired client's traffic to the DMZ cannot be made with the certain knowledge that the client is a guest.

Due to the lack of network selection for wired clients, open mode cannot be supported with guest tunneling. Open mode is when an IP address is allocated as soon as a client connects to the access switch. Once the client is connected via a tunnel, it must be reassigned an IP address from a subnet provisioned at the DMZ, before web authentication can be attempted.

Converged Guest Access Solution

Figure 3: Converged Guest Access Solution



In the preceding figure, the Converged Access device forms the attachment point for both wired and wireless sessions and provides Layer 2 authentication, where applicable. Wired session guests on a mobility agent (a foreign device) are directed through a Control And Provisioning of Wireless Access Points (CAPWAP) tunnel to the wireless controller (the anchor device) in the demilitarized zone (DMZ). The wired session guests are provided open or web-authenticated access from the wireless controller. This approach simplifies the management of guest access because only one network device is provisioned to manage HTTP traffic and serve web pages.

Tunneling wired guest traffic to the DMZ allows the same controller platform to provide web-authenticated and open access to wired guests also, further simplifying the management of guest access and ensuring a consistent experience for end users. To activate the CAPWAP tunnel, matching guest LAN profiles must be configured on foreign and anchor devices.

Authentication, authorization, and accounting (AAA) services are required at the access layer for Layer 2 authentication and, optionally, to direct the device to open a tunnel for a wired client. A DMZ uses AAA for client guest authentication. The Mobility Controller/Mobility Tunnel Endpoint (MC/MTE) allows the CAPWAP tunnel to the DMZ to be load-balanced across an array of wireless controllers.

CAPWAP Tunneling

In an enterprise Edge (eEdge) implementation of wired guest access, Control And Provisioning of Wireless Access Points (CAPWAP) tunneling is implemented as an Enterprise Policy Manager (EPM) plug-in.

When a tunnel is specified within a user profile or a service template, the EPM invokes the CAPWAP tunnel. The EPM requests that the Wireless Controller Module (WCM) establish a CAPWAP tunnel for the session on which the EPM is installed. If the WCM returns an error or indicates unsolicited tunnel termination at any subsequent point, the CAPWAP tunnel notifies the EPM of failure. The failure results in an authorization-failure event at the session manager, and a control policy rule can be specified to determine the failure handling.

The Session Manager is responsible for creating and managing wired sessions in the eEdge framework. It assigns an audit-session-id at session creation and stores client identity data in a session entry in the database.

It also manages the authentication of connecting endpoints where authentication is specified under a control policy.

Based on requests, the WCM is responsible for the CAPWAP tunneling of wired clients at an Converged Access switch. The WCM also provides identical handling of tunneled wireless and wired guest sessions at the controller.

Note A new tunnel is established only if it does not exist between the access switch and the relevant controller. If a tunnel exists, a client is added to it.

Note The Vendor-specific attribute (VSA) for activating CAPWAP tunneling using user profiles is "subscriber:capwap-tunnel-profile-name= name".

How to Configure Wired Guest Access

Configuring a Guest LAN

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **guest-lan** *profile-name* [*lan-id*]
4. **shutdown**
5. **client** {**association limit** [*max-connections*] | **vlan** [*vlan-id*]}
6. **security web-auth** [**parameter-map** *parameter-name*]
7. **mobility anchor** [*ip-address* | *mac-address*]
8. **no shutdown**
9. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**guest-lan** *profile-name* [*lan-id*] **Example:** `Device(config)# guest-lan guest-lan-name 1`	Configures the wireless guest LAN network and enters guest LAN configuration mode.
Step 4	**shutdown** **Example:** `Device(config-guest-lan)# shutdown`	Disables the guest LAN.
Step 5	**client** {**association limit** [*max-connections*] | **vlan** [*vlan-id*]} **Example:** `Device(config-guest-lan)# client vlan VLAN100`	Enables guest LAN configuration for clients.
Step 6	**security web-auth** [**parameter-map** *parameter-name*] **Example:** `Device(config-guest-lan)# security web-auth`	Configures a security policy for a guest LAN.
Step 7	**mobility anchor** [*ip-address* | *mac-address*] **Example:** `Device(config-guest-lan)# mobility anchor`	Configures mobility for a guest LAN.
Step 8	**no shutdown** **Example:** `Device(config-guest-lan)# no shutdown`	Enables the guest LAN.
Step 9	**end** **Example:** `Device(config-guest-lan)# end`	Exits guest LAN configuration mode and enters privileged EXEC mode.

Configuring a CAPWAP Tunnel in a Service Template

Perform the following task to configure a Control And Provisioning of Wireless Access Points (CAPWAP) tunnel in a service template. Perform the following task to activate a tunnel service when Layer 2 authentication failure occurs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **service-template** *template-name*
4. **tunnel type capwap name** *tunnel-name*
5. **exit**
6. **policy-map type control subscriber** *control-policy-name*
7. **event session-started** [**match-all** | **match-any**]

8. *priority-number* **class** {*control-class-name* | **always**} [**do-all** | **do-until-failure** | **do-until-success**]
9. *action-number* **authenticate using** {**dot1x** | **mab** | **webauth**}
10. **exit**
11. **exit**
12. **event authentication-failure** [**match-all** | **match-any**]
13. *priority-number* **class** {*control-class-name* | **always**} [**do-all** | **do-until-failure** | **do-until-success**]
14. *action-number* **activate** {**policy type control subscriber** *control-policy-name* | **service-template** *template-name* [**aaa-list** *list-name*] [**precedence** [**replace-all**]]}
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**service-template** *template-name* **Example:** `Device(config)# service-template GUEST-TUNNEL`	Defines a template that contains a set of service policy attributes to apply to subscriber sessions and enters service template configuration mode.
Step 4	**tunnel type capwap name** *tunnel-name* **Example:** `Device(config-service-template)# tunnel type capwap name TUNNEL-CAPWAP`	Configures a CAPWAP tunnel in a service template.
Step 5	**exit** **Example:** `Device(config-service-template)# exit`	Exits service template configuration mode and enters global configuration mode.
Step 6	**policy-map type control subscriber** *control-policy-name* **Example:** `Device(config)# policy-map type control subscriber TUNNELLED-GUEST`	Defines a control policy for subscriber sessions and enters control policy-map event configuration mode.
Step 7	**event session-started** [**match-all**	**match-any**] **Example:** `Device(config-event-control-policymap)# event session-started`	Specifies the type of event that triggers actions in a control policy if all authentication events are a match and enters control policy-map class configuration mode.

	Command or Action	Purpose
Step 8	*priority-number* **class** {*control-class-name* | **always**} [**do-all** | **do-until-failure** | **do-until-success**] **Example:** `Device(config-class-control-policymap)# 1 class always`	Specifies that the control class should execute the actions in a control policy, in the specified order, until one of the actions fails, and enters control policy-map action configuration mode.
Step 9	*action-number* **authenticate using** {**dot1x** | **mab** | **webauth**} **Example:** `Device(config-action-control-policymap)# 1 authenticate using dot1x`	Authenticates a control policy on a subscriber session.
Step 10	**exit** **Example:** `Device(config-action-control-policymap)# exit`	Exits control policy-map action configuration mode and enters control policy-map class configuration mode.
Step 11	**exit** **Example:** `Device(config-class-control-policymap)# exit`	Exits control policy-map class configuration mode and enters control policy-map event configuration mode.
Step 12	**event authentication-failure** [**match-all** | **match-any**] **Example:** `Device(config-event-control-policymap)# event authentication-failure`	Specifies the type of event that triggers actions in a control policy if all authentication events are a match and enters control policy-map class configuration mode.
Step 13	*priority-number* **class** {*control-class-name* | **always**} [**do-all** | **do-until-failure** | **do-until-success**] **Example:** `Device(config-class-control-policymap)# 1 class DOT1X-NO-RESP`	Specifies that the control class should execute the actions in a control policy, in the specified order, until one of the actions fails, and enters control policy-map action configuration mode.
Step 14	*action-number* **activate** {**policy type control subscriber** *control-policy-name* | **service-template** *template-name* [**aaa-list** *list-name*] [**precedence** [**replace-all**]]} **Example:** `Device(config-action-control-policymap)# 1 activate service-template GUEST-TUNNEL`	Activates a control policy on a subscriber session.
Step 15	**end** **Example:** `Device(config-action-control-policymap)# end`	Exits control policy-map action configuration mode and returns to privileged EXEC mode.

Configuring CAPWAP Forwarding

Perform the following task to configure a specific VLAN for CAPWAP forwarding. Once configured, this VLAN can be used only for CAPWAP forwarding.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vlan** *vlan-id*
4. **exit**
5. **access-session tunnel vlan** *vlan-id*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vlan** *vlan-id* **Example:** `Device(config)# vlan 1755`	Configures a VLAN and enters VLAN configuration mode.
Step 4	**exit** **Example:** `Device(config-vlan)# exit`	Exits VLAN configuration mode and enters global configuration mode.
Step 5	**access-session tunnel vlan** *vlan-id* **Example:** `Device(config)# access-session tunnel vlan 1755`	Configures VLAN access session to the specified tunnel. **Note** Before you use this command, configure the VLAN using the **vlan** *vlan-id* command.
Step 6	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuration Examples for Wired Guest Access

Example: Configuring a CAPWAP Tunnel in a Service Template

The following example shows how to configure a CAPWAP tunnel in a service template to enable wired guest access.

```
Device> enable
Device# configure terminal
Device(config)# service-template GUEST-TUNNEL
Device(config-service-template)# tunnel type capwap name TUNNEL-CAPWAP
Device(config-service-template)# exit
Device(config)# policy-map type control subscriber TUNNELLED-GUEST
Device(config-event-control-policymap)# event session-started
Device(config-class-control-policymap)# 1 class always
Device(config-action-control-policymap)# 1 authenticate using dot1x
Device(config-action-control-policymap)# exit
Device(config-class-control-policymap)# 1 class DOT1X-NO-RESP
Device(config-action-control-policymap)# 1 activate service-template GUEST-TUNNEL
Device(config-action-control-policymap)# end
```

Example: Configuring the Mobility Agent

The following example shows how to configure interface ports on the mobility agent (anchor).

Wired-guest-access ports initially terminate on a Layer 2 access switch or switch port configured with VLAN interfaces for wired-guest-access traffic. The wired guest traffic is then trunked from the access switch to a controller. This controller is configured with an interface that is mapped to a wired-guest-access VLAN on the access switch.

```
!
interface GigabitEthernet1/0/44
 description Connected to Client_Laptop
 switchport access vlan 10
 switchport mode access
 access-session host-mode single-host
 access-session closed
 access-session port-control auto
 access-session control-direction in
 mab
 dot1x pae authenticator
 dot1x timeout tx-period 5
 service-policy type control subscriber Guest-Access
!
interface GigabitEthernet1/0/1
 description Connected_to_MobilityController
 switchport mode trunk
!
interface Vlan10
 description CLIENT-VLAN
 ip address 172.16.10.201 255.255.255.0
 ip helper-address 172.16.10.200
!
interface Vlan80
 description MANAGEMENT-VLAN
```

```
 ip address 10.20.1.1 255.255.255.0
!
wireless management interface Vlan80
wireless mobility controller ip 10.20.1.2 public-ip 10.20.1.2 << Mobility Controller IP >>
!
guest-lan glan-1 1
 shutdown
 client vlan Vlan10
 no security web-auth << Use "security webauth" for webauth access & "no security webauth"
 for open access. >>
 mobility anchor 10.20.1.3 << Guest Controller IP >>
 no shutdown
!
```

Example: Configuring the Mobility Controller

The following example shows how to configure the interface ports and wireless mobility on the mobility controller to enable wired guest access.

```
!
interface TenGigabitEthernet1/0/2
 description Connected-to-MobilityAgent
 switchport mode trunk
!
interface TenGigabitEthernet1/0/1
 description Connected-to-GuestController
 switchport mode trunk
!
interface Vlan80
 description MANAGEMENT-VLAN
 ip address 10.20.1.2 255.255.255.0
!
wireless management interface Vlan80
!
wireless mobility controller peer-group pg-name
wireless mobility controller peer-group pg-name member ip 10.20.1.1 public-ip 10.20.1.1 <<
 Mobility Agent IP >>
!
wireless mobility group member ip 10.20.1.3 public-ip 10.20.1.3 << Guest Controller IP >>
wireless mobility group name mcg-name
!
```

Example: Configuring the Guest Controller

The following example shows how to configure interface ports on the guest controller (anchor) and how to set up DHCP snooping.

The guest (local WLAN) controller anchors the client onto a demilitarized zone (DMZ) anchor WLAN controller that is configured for wired and wireless guest access. After a successful handoff of the client to the DMZ anchor controller, the DHCP IP address assignment, client authentication, and so on are handled in the DMZ Cisco Wireless LAN Controller (WLC). After WLC completes the authentication, the client is allowed to send and receive traffic.

```
!
interface TenGigabitEthernet1/0/1
 description Connected_to_MC
 switchport mode trunk
```

```
!
interface Vlan10
 description CLIENT-VLAN
 ip address 172.16.10.200 255.255.255.0
!
interface Vlan80
 description MANAGEMENT-VLAN
 ip address 10.20.1.3 255.255.255.0
!
ip dhcp snooping vlan 10
ip dhcp snooping
ip dhcp excluded-address 172.16.10.100 172.16.10.255
ip dhcp pool vlan10
 network 172.16.10.0 255.255.255.0
 default-router 172.16.10.200
!
wireless management interface Vlan80
!
wireless mobility group name mcg-name
wireless mobility group member ip 10.20.1.2 public-ip 10.20.1.2 << Mobility Controller IP
>>
!
guest-lan glan-1 1
 shutdown
 client vlan Vlan10
 no security web-auth << Use "security web-auth" for web-auth access & "no security web-auth"
 for open access. >>
 mobility anchor
 no shutdown
!
```

Example: Configuring CAPWAP Forwarding

```
Device> enable
Device# configure terminal
Device(config)# vlan 1755
Device(config-vlan)# exit
Device(config)# access-session tunnel vlan 1775
Device(config)# end
```

Additional References for Wired Guest Access

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco Identity-Based Networking Services commands	Cisco IOS Identity-Based Networking Services Command Reference

Standards and RFCs

RFC	Title
IEEE 802.1X	*Port based Network Access Control*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

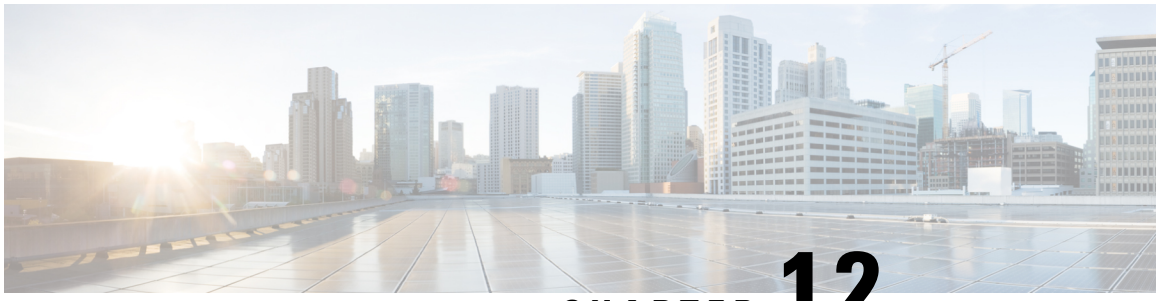
Feature Information for Wired Guest Access

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

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

Table 12: Feature Information for Wired Guest Access

Feature Name	Releases	Feature Information
Wired Guest Access	Cisco IOS XE Release 3.3SE	The Wired Guest Access feature enables guest users of an enterprise network, that supports both wired and wireless access, to connect to the guest access network from a wired Ethernet connection. The wired Ethernet connection is designated and configured for guest access. Wired session guests on mobility agents are directed to a wireless guest controller in a demilitarized zone (DMZ) through a Control And Provisioning of Wireless Access Points (CAPWAP) tunnel. In Cisco IOS XE Release 3.6E, this feature is supported on the following platforms: • Cisco 5700 Series Wireless LAN Controllers • Cisco Catalyst 3850 Series Switches • Cisco Catalyst 3650 Series Switches The following commands were introduced or modified: **access-session tunnel vlan**, **event**, **match authorization-failure**, **tunnel type capwap**.



CHAPTER 12

Web Authentication Redirection to Original URL

The Web Authentication Redirection to Original URL feature enables networks to redirect guest users to the URL that they had originally requested. This feature is enabled by default and requires no configuration. This module provides information about this feature.

Information About Web Authentication Redirection to Original URL

Web Authentication Redirection to Original URL Overview

The Web Authentication Redirection to Original URL feature enables networks to redirect guest users to the URL that they had originally requested. This feature is enabled by default and requires no configuration.

Guest networks are network connections provided by an enterprise to allow their guests to gain access to the Internet and to their own enterprise networks without compromising the security of the host enterprise. Guest users of an enterprise network can connect to the guest access network through either a wired Ethernet connection or a wireless connection.

Guest access uses a captive portal to gather all web requests made by guests and redirect these requests to one of the guest on-boarding web pages. When guests successfully complete the guest workflow, they are redirected to the page that they had originally requested.

The originally requested URL is passed as metadata along with the Cisco Identity Services Engine (ISE) guest access redirect URL. The Cisco ISE is a security policy management and control platform. It automates and simplifies access control and security compliance for wired, wireless, and VPN connectivity. The requested URL is added at the end of the Cisco ISE guest URL so that the device can send the redirect URL to the guest client. The Cisco ISE parses the URL and redirects the guest to the original URL after completing the on-boarding.

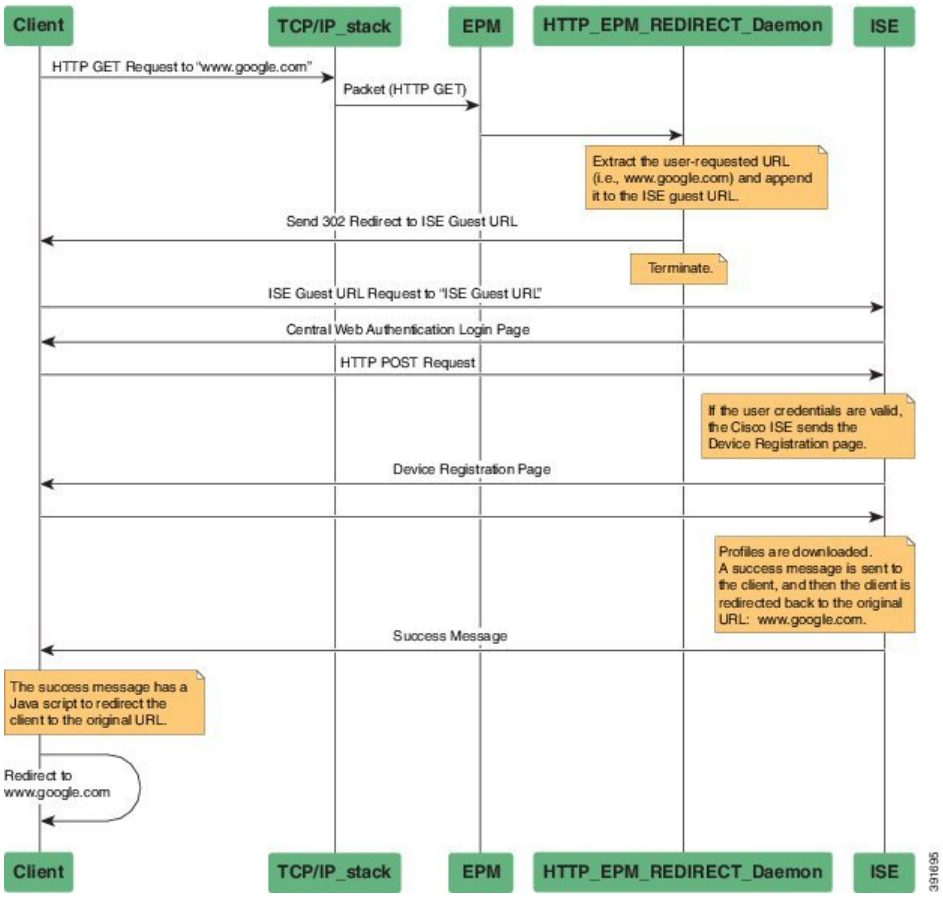
The following is an example of a redirect URL along with the original requested URL:

```
https://10.64.67.92:8443/guestportal/gateway?sessionId=0920269E0000000B0002426B&action=cwa&redirect_
url=http://www.cisco.com/
```

In this example, the URL, https://10.64.67.92:8443/guestportal/gateway?sessionId=0920269E0000000B0002426B&action=cwa is the URL for the guest portal, "&" tells the browser that what follows is a list of name value pairs, and redirect_url=http://www.cisco.com identifies the URL that the user originally requested and to which the user is redirected after completing the guest workflow.

This illustration displays the packet flow that redirects a user to the originally requested URL:

Figure 4: Original URL Redirection Packet Flow



1. A user accesses a network for the first time and sends an HTTP request to access www.google.com. When the user first accesses the network, a MAC authentication bypass (MAB) is triggered and the MAC address is sent to the Cisco ISE.
2. The Cisco ISE returns a RADIUS access-accept message (even if the MAC address is not received) along with the redirect access control list (ACL), the ACL-WEBAUTH-REDIRECT message, and the guest web portal URL to the device.

   The RADIUS message instructs the device to open a port that is restricted based on the configured port and the redirect ACLs, for regular network traffic.
3. When the user launches a web browser, the device intercepts the HTTP traffic and redirects the browser to the Cisco ISE central web authentication (CWA) guest web portal URL; the user-requested URL is extracted and appended to the Cisco ISE guest URL.

4. When the user is authenticated, the Cisco ISE sends the Device Registration page to the user. The user enters the required information, and the page is returned to the Cisco ISE. The Cisco ISE downloads user profiles and redirects the user to the originally requested URL: www.google.com.

Additional References for Web Authentication Redirection to Original URL

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IBNS commands	Cisco IOS Identity-Based Networking Services Command Reference
Wired guest access	"Wired Guest Access" module of the *Identity-Based Networking Services Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Web Authentication Redirection to Originial URL

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

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

Feature Name	Releases	Feature Information
Web Authentication Redirection to Original URL	Cisco IOS XE 3.6.0E	The Web Authentication Redirection to Original URL feature enables networks to redirect guest users to the original URL that they had request. This feature is enabled by default and requires no configuration. In Cisco IOS XE Release 3.6.0E, this feature is supported on the following platforms: • Cisco Catalyst 3850 Series Switches • Cisco Catalyst 3650 Series Switches No commands were added or updated for this feature.