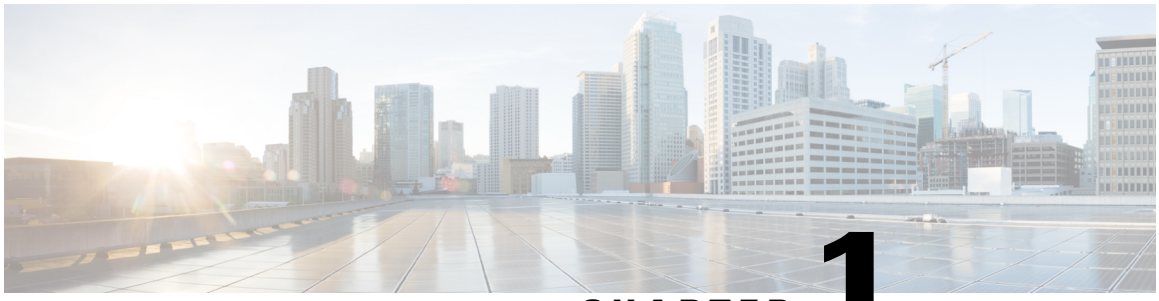
Identity-Based Networking Services Configuration Guide, Cisco IOS XE Release 3SE (Catalyst 3650 Switches)

First Published: January 29, 2013

Last Modified: January 29, 2013

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 at the end of this module.

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.

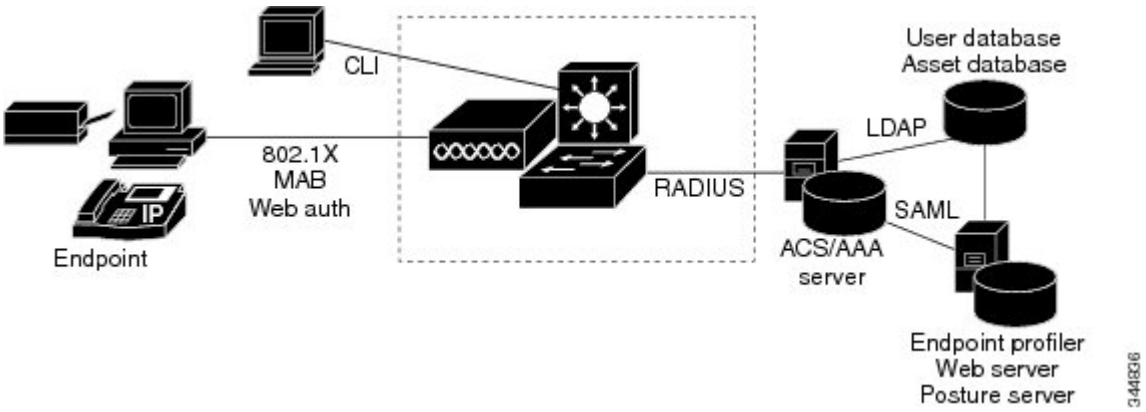
Information About Identity-Based Networking Services

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

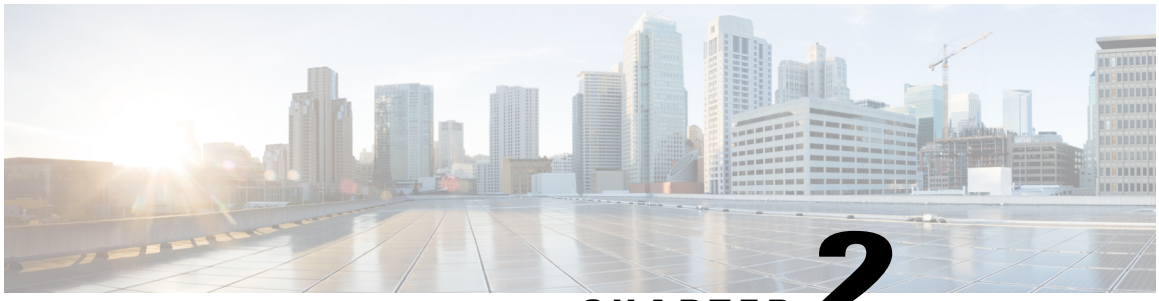
Feature Information for Identity-Based Networking Services Overview

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	Allows a single session identifier to be used for all web authentication sessions in addition to 802.1X and MAB authenticated sessions.



CHAPTER 2

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 at the end of this module.

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

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 8" 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.

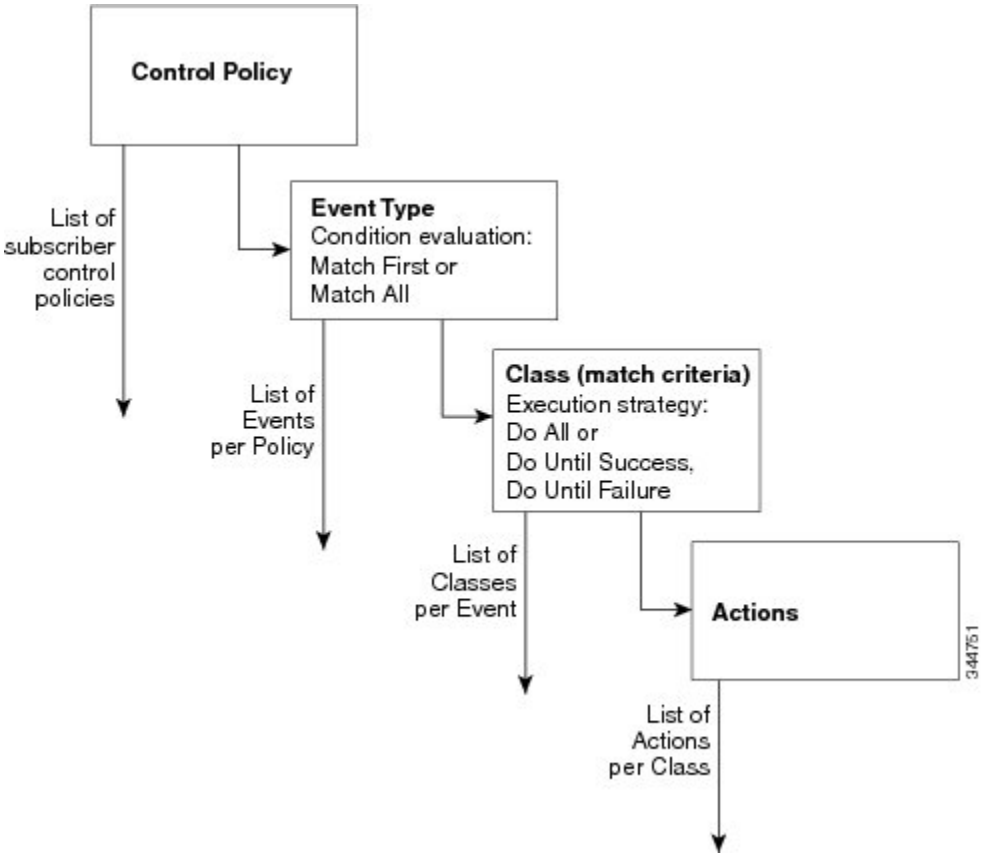
Control Policies for Identity-Based Networking Services

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 31" 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. If you boot the standby device while in new-style mode, the standby device will be in new-style mode and after switchover the device remains in new-style mode. To switch back to legacy mode, you must use the **authentication display legacy** command and reload the standby switch.

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.

	Command or Action	Purpose
		• **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.
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.

	Command or Action	Purpose
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:** `Device(config-filter-control-classmap)# match method dot1x`	(Optional) Creates a condition that evaluates true based on an event's authentication method.
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.

	Command or Action	Purpose
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.

	Command or Action	Purpose
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:** `Device(config-action-control-policymap)# 10 authenticate using dot1x priority 10`	(Optional) Initiates the authentication of a subscriber session using the specified method.
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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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 24" 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 tengigabitethernet 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 TenGigabitEthernet 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**}	Allows hosts to gain access to a controlled port.

	Command or Action	Purpose
	Example: `Device(config-if)# access-session host-mode single-host`	• 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.
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 email**
6. **custom-page** {**failure** | **login [expired]** | **success**} **device** *location*:*filename*
7. **max-http-conns** *number*
8. **ratelimit init-state-sessions** *rate-limit*
9. **redirect** {{**for-login** | **on-failure** | **on-success**} *url* | **portal** {**ipv4** *ipv4-address* | **ipv6** *ipv6-address*}}
10. **timeout init-state min** *minutes*
11. **type** {**authbypass** | **consent** | **webauth** | **webconsent**}
12. **virtual-ip** {**ipv4** *ipv4-address* | **ipv6** *ipv6-address*}
13. **watch-list** {**add-item** {**ipv4** *ipv4-address* | **ipv6** *ipv6-address*} | **dynamic-expiry-timeout** *minutes* | **enabled**}
14. **end**
15. **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.

	Command or Action	Purpose
Step 5	**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 6	**custom-page** {**failure** | **login [expired]** | **success**} **device** *location*:*filename*
	
	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`
		(Optional) Displays custom authentication proxy web pages during web-based authentication.
		• You must configure all four custom HTML files. If fewer than four files are configured, the internal default HTML pages will be used.

Step 7	**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 8	**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 9	**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 10	**timeout init-state min** *minutes* **Example:** `Device(config-params-parameter-map)# timeout init-state min 15`	(Optional) Sets the Init state timeout for web-based authentication sessions.
Step 11	**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.

	Command or Action	Purpose
Step 12	**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 13	**watch-list** {**add-item** {**ipv4** *ipv4-address* | **ipv6** *ipv6-address*} | **dynamic-expiry-timeout** *minutes* | **enabled**}
	
	Example:
	`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`
		(Optional) Enables a watch list of web-based authentication clients.
		• This command is supported in the global parameter map only.

Step 14	**end** **Example:** `Device(config-params-parameter-map)# end`	(Optional) Exits parameter-map configuration mode and returns to privileged EXEC mode.
Step 15	**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 min 15
 max-http-conns 5
 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 13" 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

Related Topic	Document Title
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

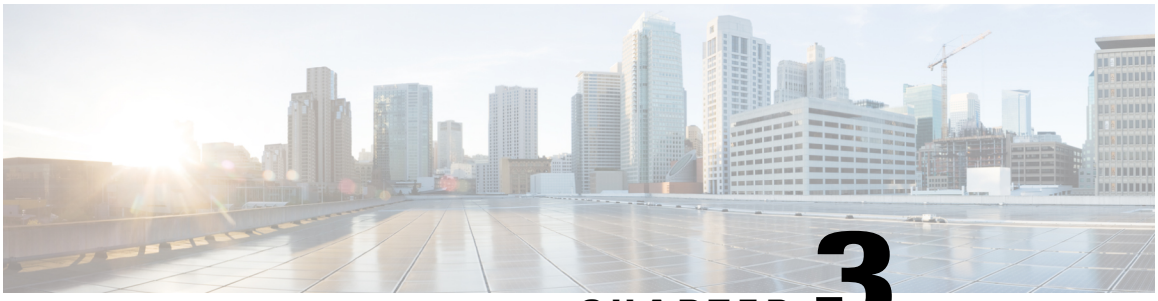
Feature Information for Identity Control Policies

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 2: 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. 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**.
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.
Per User Inactivity Handling across Methods	Cisco IOS XE Release 3.2SE	Supports RADIUS attributes 28 (Idle-Timeout) and 29 (Termination-Action).



CHAPTER 3

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 at the end of this module.

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. **tag** *tag-name*
10. **vlan** *vlan-id*
11. **end**
12. **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	**tag** *tag-name* **Example:** `Device(config-service-template)# tag TAG_2`	(Optional) Associates a user-defined tag with a service template.
Step 10	**vlan** *vlan-id* **Example:** `Device(config-service-template)# vlan 215`	(Optional) Applies a VLAN to sessions using a service template.
Step 11	**end** **Example:** `Device(config-service-template)# end`	Exits service template configuration mode and returns to privileged EXEC mode.
Step 12	**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 13."

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*
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

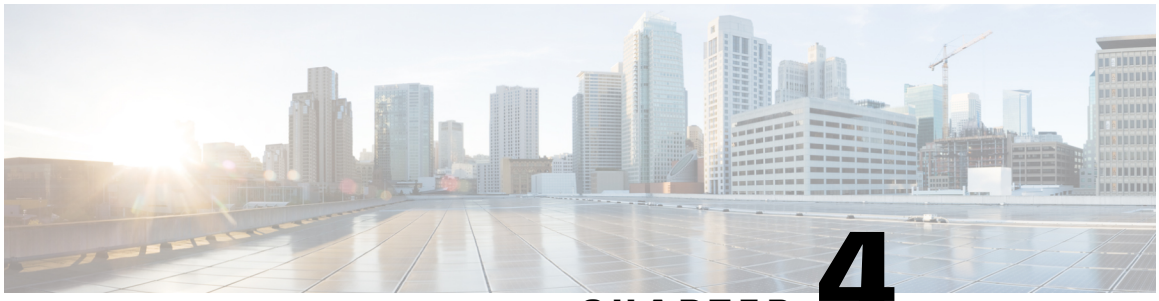
Feature Information for Identity Service Templates

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 3: 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.
Identity Service Template	Cisco IOS XE Release 3.2SE	Enables identity service templates to be configured locally and available at all times. 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 4

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 at the end of this module.

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 4: 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.

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 40" 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 40" 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 40" 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 40" 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 40" 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

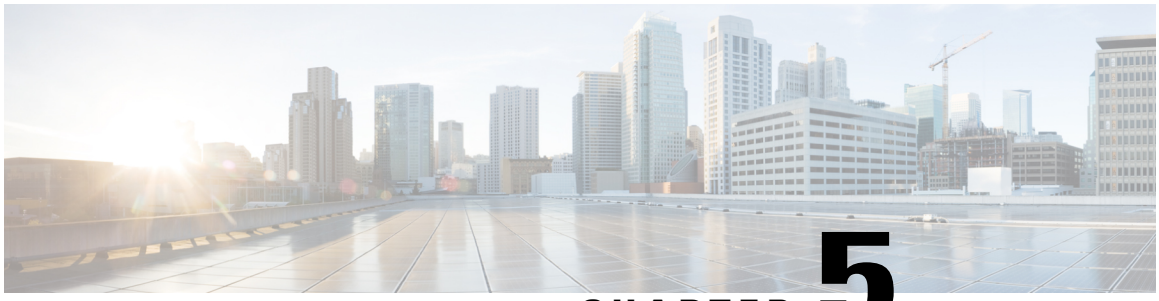
Feature Information for CoA Support

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 5: 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.



CHAPTER 5

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 at the end of this module.

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 local authentication** {*method-list-name* | **default**} **authorization** {*method-list-name* | **default**}
4. **username** *name* **aaa attribute list** *aaa-attribute-list* [**password** *password*]
5. **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 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 4	**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 5	**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 group server radius** *group-name*
4. **subscriber mac-filtering security-mode** {**mac** | **none** | **shared-secret**}
5. **mac-delimiter** {**colon** | **hyphen** | **none** | **single-hyphen**}
6. **exit**
7. **username** *mac-address* **mac** [**aaa attribute list** *aaa-attribute-list*]
8. **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.

	Command or Action	Purpose
Step 3	**aaa group server radius** *group-name* **Example:** `Device(config)# aaa group server radius RAD_GROUP1`	Groups different RADIUS server hosts into distinct lists.
Step 4	**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 5	**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 6	**exit** **Example:** `Device(config-sg-radius)# exit`	Exits server group configuration mode and returns to global configuration mode.
Step 7	**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 8	**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 group server radius** *group-name*
5. **server** *ip-address* [**auth-port** *port-number*] [**acct-port** *port-number*]
6. **key-wrap enable**
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.
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 group server radius** *group-name* **Example:** `Device(config)# aaa group server radius RAD_GROUP1`	Groups different RADIUS server hosts into distinct lists.
Step 5	**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 6	**key-wrap enable** **Example:** `Device(config-sg-radius)# key-wrap enable`	Enables AES key wrap for this RADIUS server.

	Command or Action	Purpose
Step 7	**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 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 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

Related Topic	Document Title
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

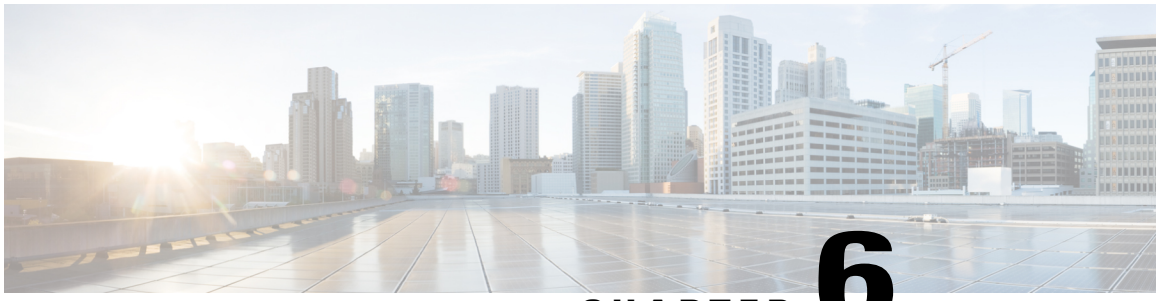
Feature Information for Local Authentication Using LDAP

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 6: 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). 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 6

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 at the end of this module.

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.

	Command or Action	Purpose
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:** `Device(config-service-template)# vlan 116`	Assigns a VLAN to a subscriber session.
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.

	Command or Action	Purpose
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.
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** **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 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**]	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.

	Command or Action	Purpose
	Example: `Device(config-class-control-policymap)# 10 class AAA-SVR-DOWN-UNAUTHD-HOST do-until-failure`	
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.
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

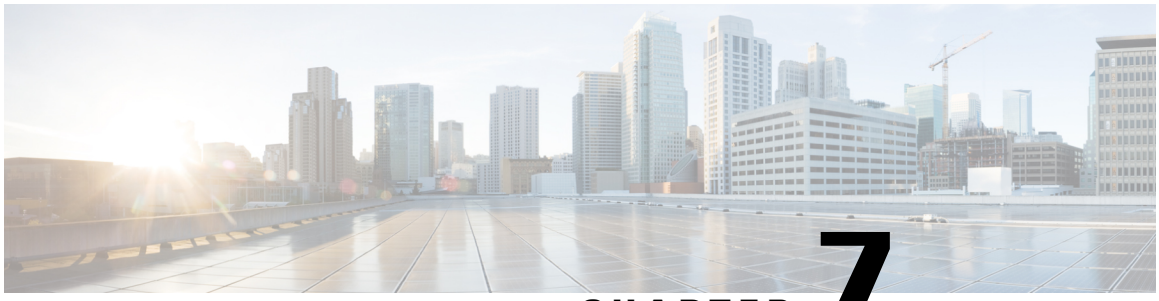
Feature Information for Critical Voice VLAN Support

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 7: Feature Information for Critical Voice VLAN Support

Feature Name	Releases	Feature Information
Critical Voice VLAN Support	15.2(1)E 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. The following command was added or modified: **voice vlan**



CHAPTER 7

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 standalone configuration or in a dual-controller configuration that uses both an anchor controller and a foreign controller. A dual-controller configuration isolates wired guest access traffic; however it is not required for deployment of the wired guest access.

Wired-guest-access ports initially terminate on a Layer 2 access switch or switch port that is 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 aWired-guest-access VLAN on the access switch.

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 at the end of this module.

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 Cisco Next Generation Wiring Closet (NGWC) device that is configured for open mode.
- Tunneling of wired clients is not supported after successful web authentication at the NGWC device because automated IP address reassignment is not supported after web-authentication.
- The NGWC 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

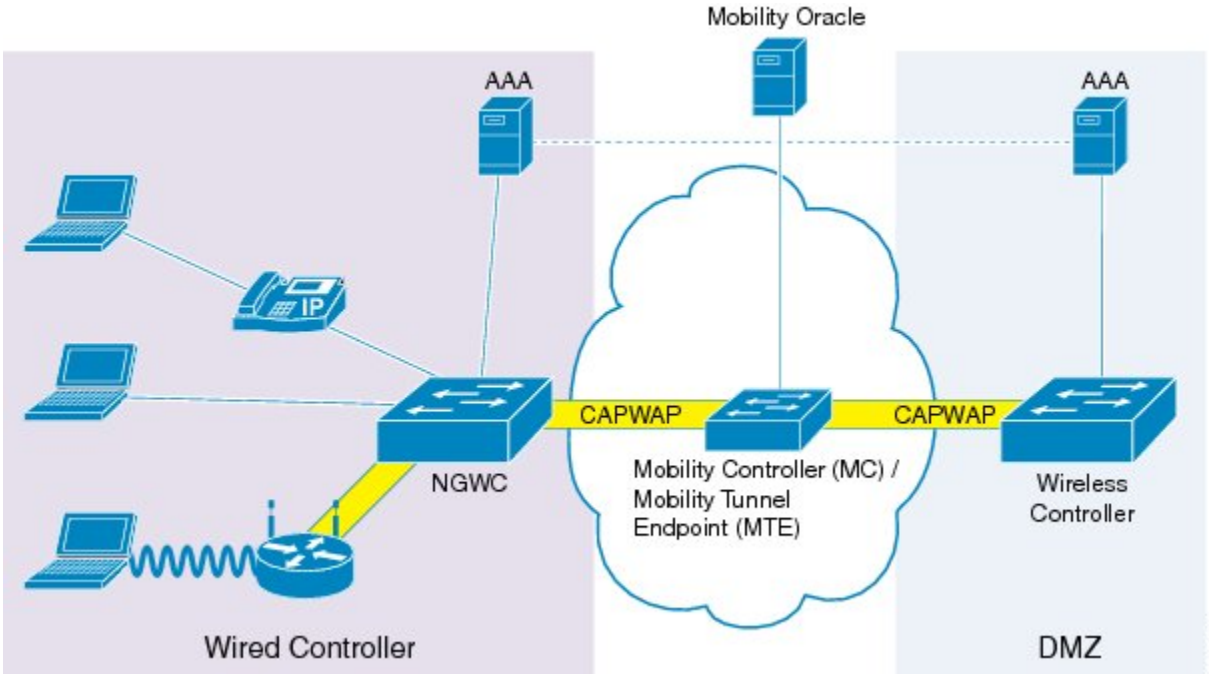
Enterprise networks that support both wired and wireless access need to provide guest services that are consistent across the two access media, from a perspective of both client experience and manageability. For wireless networks, guest traffic from a mobility anchor device is directed typically through 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 Cisco Next Generation Wiring Closet (NGWC) 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 Cisco Next Generation Wiring Closet (NGWC) 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 NGWC 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.
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.

	Command or Action	Purpose
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 seesion-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.
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**]	Specifies that the control class should execute the actions in a control policy, in the specified order, until one of the

	Command or Action	Purpose
	Example: `Device(config-class-control-policymap)# 1 class DOT1X-NO-RESP`	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.

	Command or Action	Purpose
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 seesion-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 8: 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. The following commands were introduced or modified: **access-session tunnel vlan**, **event**, **match authorization-failure**, **tunnel type capwap**.