CHAPTER 49

# Configuring Wired Guest Access

This chapter describes the tasks for configuring wired guest access on a Catalyst 4500 series switch and includes these major sections:

**Note** For complete syntax and usage information for the switch commands used in this chapter, see the *Cisco Catalyst 4500 Series Switch Command Reference* and related publications at this location: http://www.cisco.com/en/US/products/hw/switches/ps4324/index.html

If a command is not in the *Catalyst 4500 Series Switch Command Reference*, you can locate it in the Cisco IOS library. See related publications at this location: http://www.cisco.com/en/US/products/ps6350/index.html

**Note** To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release.

## Wired Guest Access

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

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

### Restrictions for Wired Guest Access

- Wired guest access does not work if host is connected on the VSS standby switch port.
- Wired guest access does not work on Supervisor Engine 8-E, in multiple-host mode or in multi-authentication mode.
- Wired guest access works on Supervisor Engine 8-E in wireless mode only.

- 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

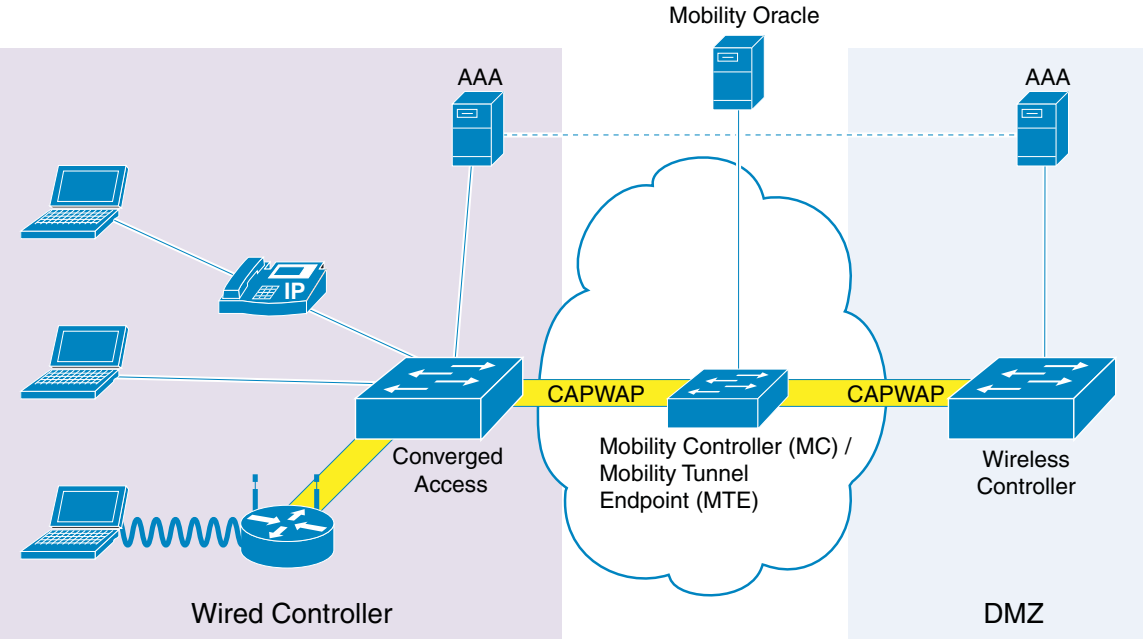
## Wired Guest Access Overview

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




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 Guest LAN

To configure a guest LAN, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | `Switch>enable` | Enables privileged EXEC mode. Enter your password if prompted. |
| **Step 2** | `Switch#configure terminal` | Enters global configuration mode. |
| **Step 3** | `Switch(config)#guest-lan profile-name [lan-id]` | Configures the wireless guest LAN network and enters guest LAN configuration mode. |
| **Step 4** | `Switch(config-guest-lan)#shutdown` | Disables the guest LAN. |
| **Step 5** | `Switch(config-guest-lan)#client {association limit [max-connections] \| vlan [vlan-id]}` | Enables guest LAN configuration for clients. |
| **Step 6** | `Switch(config-guest-lan)#security web-auth [parameter-map parameter-name]` | Configures a security policy for a guest LAN. |
| **Step 7** | `Switch(config-guest-lan)#mobility anchor [ip-address \| mac-addressI]` | Configures mobility for a guest LAN. |
| **Step 8** | `Switch(config-guest-lan)#no shutdown` | Enables the guest LAN. |
| **Step 9** | `Switch(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 and to activate a tunnel service when Layer 2 authentication failure occurs.

| | Command | Purpose |
|---|---|---|
| **Step 1** | `Switch>enable` | Enables privileged EXEC mode. Enter your password if prompted. |
| **Step 2** | `Switch#configure terminal` | Enters global configuration mode. |
| **Step 3** | `Switch (config)# service-template` *template-name* | Defines a template that contains a set of service policy attributes to apply to subscriber sessions and enters service template configuration mode. |
| **Step 4** | `Switch (config-service-template)# tunnel type capwap name` *tunnel-name* | Configures a CAPWAP tunnel in a service template. |
| **Step 5** | `Switch (config-service-template)# exit` | Exits service template configuration mode and enters global configuration mode. |
| **Step 6** | `Switch (config)# policy-map type control subscriber` *control-policy-name* | Defines a control policy for subscriber sessions and enters control policy-map event configuration mode. |
| **Step 7** | `Switch (config-event-control-policymap)# event session-started` [**match-all** \| **match-any**] | 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** | `Switch (config-class-control-policymap)#` *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. |
| **Step 9** | `Switch (config-action-control-policymap)#` *action-number* **authenticate using** {**dot1x** \| **mab** \| **webauth**} | Authenticates a control policy on a subscriber session. |
| **Step 10** | `Switch (config-action-control-policymap)# exit` | Exits control policy-map action configuration mode and enters control policy-map class configuration mode. |
| **Step 11** | `Switch (config-class-control-policymap)# exit` | Exits control policy-map class configuration mode and enters control policy-map event configuration mode. |
| **Step 12** | `Switch (config-event-control-policymap)# event authentication-failure` [**match-all** \| **match-any**] | 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** | `Switch (config-class-control-policymap)#` *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. |
| **Step 14** | `Switch (config-action-control-policymap)#` *action-number* **activate** {**policy type control subscriber** *control-policy-name* \| **service-template** *template-name* [**aaa-list** *list-name*] [**precedence** [**replace-all**]]} | Activates a control policy on a subscriber session. |
| **Step 15** | `Switch (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.

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

```
Switch> enable
Switch# configure terminal
Switch(config)# service-template GUEST-TUNNEL
Switch(config-service-template)# tunnel type capwap name TUNNEL-CAPWAP
Switch(config-service-template)# exit
Switch(config)# policy-map type control subscriber TUNNELLED-GUEST
Switch(config-event-control-policymap)# event seesion-started
Switch(config-class-control-policymap)# 1 class always
Switch(config-action-control-policymap)# 1 authenticate using dot1x
Switch(config-action-control-policymap)# exit
Switch(config-class-control-policymap)# 1 class DOT1X-NO-RESP
Switch(config-action-control-policymap)# 1 activate service-template GUEST-TUNNEL
Switch(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 GigabitEthernet1/1
 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 GigabitEthernet1/2
 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 GigabitEthernet1/1
 description Connected-to-MobilityAgent
 switchport mode trunk
!
interface GigabitEthernet1/2
 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 GigabitEthernet1/1
 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

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