Web-Based Authentication

This chapter describes how to configure web-based authentication on the device. It contains these sections:

Local Web Authentication Overview

Web authentication is a Layer 3 security solution designed for providing easy and secure guest access to hosts on WLAN with open authentication or appropriate layer 2 security methods. Web authentication allows users to get authenticated through a web browser on a wireless client, with minimal configuration on the client side. It allows users to associate with an open SSID without having to set up a user profile. The host receives an IP address and DNS information from the DHCP server, however cannot access any of the network resources until they authenticate successfully. When the host connects to the guest network, the WLC redirects the host to an authentication web page where the user needs to enter valid credentials. The credentials are authenticated by the WLC or an external authentication server and if authenticated successfully is given full access to the network. Hosts can also be given limited access to particular network resources before authentication for which the pre-authentication ACL functionality needs to be configured.

The following are the different types of web authentication methods:

- Local Web Authentication (LWA): Configured as Layer 3 security on the controller, the web authentication page and the pre-authentication ACL are locally configured on the controller. The controller intercepts htttp(s) traffic and redirects the client to the internal web page for authentication. The credentials entered by the client on the login page is authenticated by the controller locally or through a RADIUS or LDAP server.

- External Web Authentication (EWA): Configured as Layer 3 security on the controller, the controller intercepts htttp(s) traffic and redirects the client to the login page hosted on the external web server. The credentials entered by the client on the login page is authenticated by the controller locally or through a RADIUS or LDAP server. The pre-authentication ACL is configured statically on the controller.

- Central Web Authentication (CWA): Configured mostly as Layer 2 security on the controller, the redirection URL and the pre-authentication ACL reside on ISE and are pushed during layer 2 authentication

to the controller. The controller redirects all web traffic from the client to the ISE login page. ISE validates the credentials entered by the client through HTTPS and authenticates the user.

Use the local web authentication feature, known as web authentication proxy, to authenticate end users on host systems that do not run the IEEE 802.1x supplicant.

When a client initiates an HTTP session, local web authentication intercepts ingress HTTP packets from the host and sends an HTML login page to the users. The users enter their credentials, which the local web authentication feature sends to the authentication, authorization, and accounting (AAA) server for authentication.

If authentication succeeds, local web authentication sends a Login-Successful HTML page to the host and applies the access policies returned by the AAA server.

If authentication fails, local web authentication forwards a Login-Fail HTML page to the user, prompting the user to retry the login. If the user exceeds the maximum number of attempts, local web authentication forwards a Login-Expired HTML page to the host, and the user is excluded with the exclusion reason as Web authentication failure.

When a client reaches maximum HTTP connections (maximum of 200 connections when configured), it will cause Transmission Control Protocol (TCP) resets and client exclusion.

Note You should use either global or named parameter-map under WLAN (for method-type, custom, and redirect) for using the same web authentication methods, such as consent, web consent, and webauth. Global parameter-map is applied by default, if none of the parameter-map is configured under WLAN.

Note The traceback that you receive when webauth client tries to do authentication does not have any performance or behavioral impact. It happens rarely when the context for which FFM replied back to EPM for ACL application is already dequeued (possibly due to timer expiry) and the session becomes ‘unauthorized’.

Note When command authorization is enabled as a part of AAA Authorization configuration through TACACS and the corresponding method list is not configured as a part of the HTTP configuration, WebUI pages will not load any data. However, some wireless feature pages may work as they are privilege based and not command based.

Based on where the web pages are hosted, the local web authentication can be categorized as follows:

- *Internal*—The internal default HTML pages (Login, Success, Fail, and Expire) in the controller are used during the local web authentication.
- *Customized*—The customized web pages (Login, Success, Fail, and Expire) are downloaded onto the controller and used during the local web authentication.
- *External*—The customized web pages are hosted on the external web server instead of using the in-built or custom web pages.

Based on the various web authentication pages, the types of web authentication are as follows:

- *Webauth*—This is a basic web authentication. Herein, the controller presents a policy page with the user name and password. You need to enter the correct credentials to access the network.
- *Consent or web-passthrough*—Herein, the controller presents a policy page with the Accept or Deny buttons. You need to click the Accept button to access the network.
- *Webconsent*—This is a combination of webauth and consent web authentication types. Herein, the controller presents a policy page with Accept or Deny buttons along with user name or password. You need to enter the correct credentials and click the Accept button to access the network.

Note

- You can view the webauth parameter-map information using the **show running-config** command output.
- The wireless Web-Authentication feature does not support the bypass type.
- Change in web authentication parameter map redirect login URL does not occur until a AP rejoin happens. You must enable and disable the WLAN to apply the new URL redirection.

Note We recommend that you follow the Cisco guidelines to create a customized web authentication login page. If you have upgraded to the latest versions of Google Chrome or Mozilla Firefox browsers, ensure that your webauth bundle has the following line in the *login.html* file:

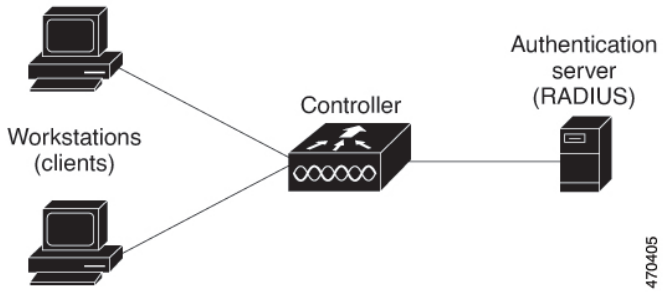
```
<body onload="loadAction();">
```

Device Roles

With local web authentication, the devices in the network have these specific roles:

- *Client*—The device (workstation) that requests access to the network and the controller and responds to requests from the controller. The workstation must be running an HTML browser with Java Script enabled.
- *Authentication server*—Authenticates the client. The authentication server validates the identity of the client and notifies the controller that the client is authorized to access the network and the controller services or that the client is denied.
- *Controller*—Controls the physical access to the network based on the authentication status of the client. The controller acts as an intermediary (proxy) between the client and the authentication server, requesting identity information from the client, verifying that information with the authentication server, and relaying a response to the client.

Figure 1: Local Web Authentication Device Roles



Authentication Process

When the page is hosted on the controller, the controller uses its virtual IP (a non-routable IP like 192.0.2.1 typically) to serve the request. If the page is hosted externally, the web redirection sends the client first to the virtual IP, which then sends the user again to the external login page while it adds arguments to the URL, such as the location of the virtual IP. Even when the page is hosted externally, the user submits its credentials to the virtual IP.

When you enable local web authentication, these events occur:

- The user initiates an HTTP session.
- The HTTP traffic is intercepted, and authorization is initiated. The controller sends the login page to the user. The user enters a username and password, and the controller sends the entries to the authentication server.
- If the authentication succeeds, the controller downloads and activates the user's access policy from the authentication server. The login success page is sent to the user.
- If the authentication fails, the controller sends the login fail page. The user retries the login. If the maximum number of attempts fails, the controller sends the login expired page, and the host is placed in a watch list. After the watch list times out, the user can retry the authentication process.
- If authentication server is not available, after the web authentication retries, the client moves to the excluded state and the client receives an Authentication Server is Unavailable page.
- The controller reauthenticates a client when the host does not respond to an ARP probe on a Layer 2 interface, or when the host does not send any traffic within the idle timeout on a Layer 3 interface.
- Web authentication sessions can not apply new VLAN as part of the authorization policy, as the client already has been assigned an IP address and you will not be able to change the IP address in the client, in case the VLAN changes.
- If the terminate action is default, the session is dismantled, and the applied policy is removed.

Note Do not use semicolons (;) while configuring username for GUI access.

Local Web Authentication Banner

With Web Authentication, you can create a default and customized web-browser banners that appears when you log in to the controller.

The banner appears on both the login page and the authentication-result pop-up pages. The default banner messages are as follows:

- *Authentication Successful*
- *Authentication Failed*
- *Authentication Expired*

The Local Web Authentication Banner can be configured as follows:

- Use the following global configuration command:

```
Device(config)# parameter map type webauth global
Device(config-params-parameter-map)# banner ?
file <file-name>
text <Banner text>
title <Banner title>
```

The default banner *Cisco Systems* and *Switch host-name Authentication* appear on the Login Page. *Cisco Systems* appears on the authentication result pop-up page.

Figure 2: Authentication Successful Banner

The banner can be customized as follows:

- Add a message, such as switch, router, or company name to the banner:

- New-style mode—Use the following global configuration command:

  parameter-map type webauth global

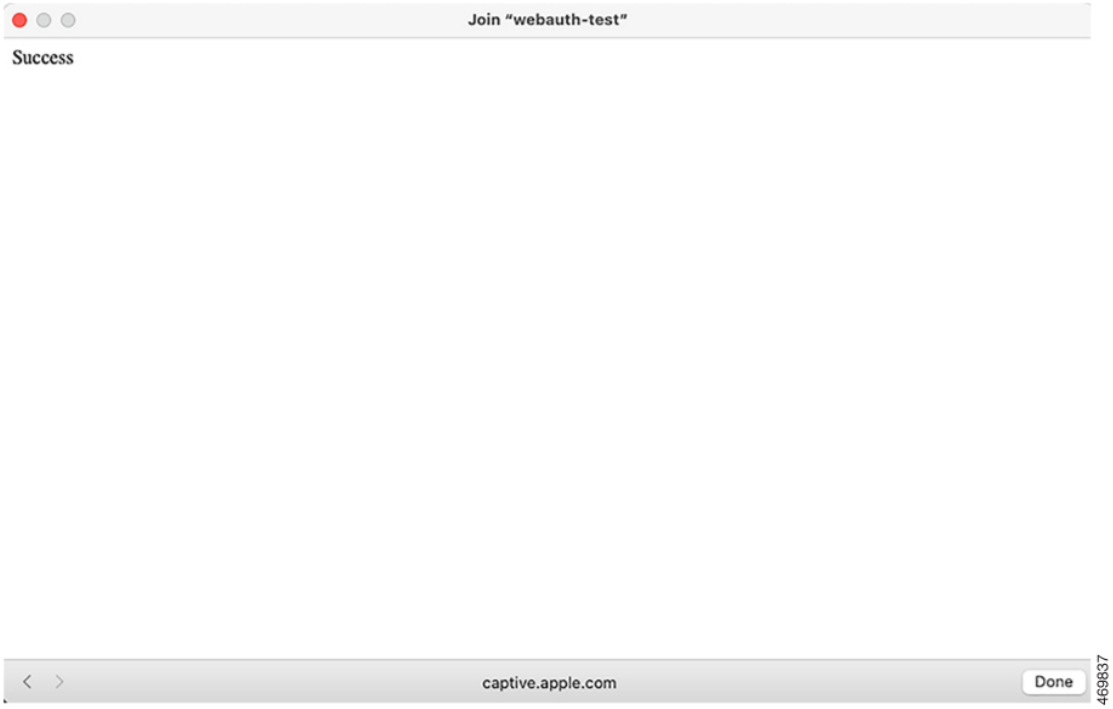
  banner text <text>

- Add a logo or text file to the banner:
  - New-style mode—Use the following global configuration command:

    parameter-map type webauth global

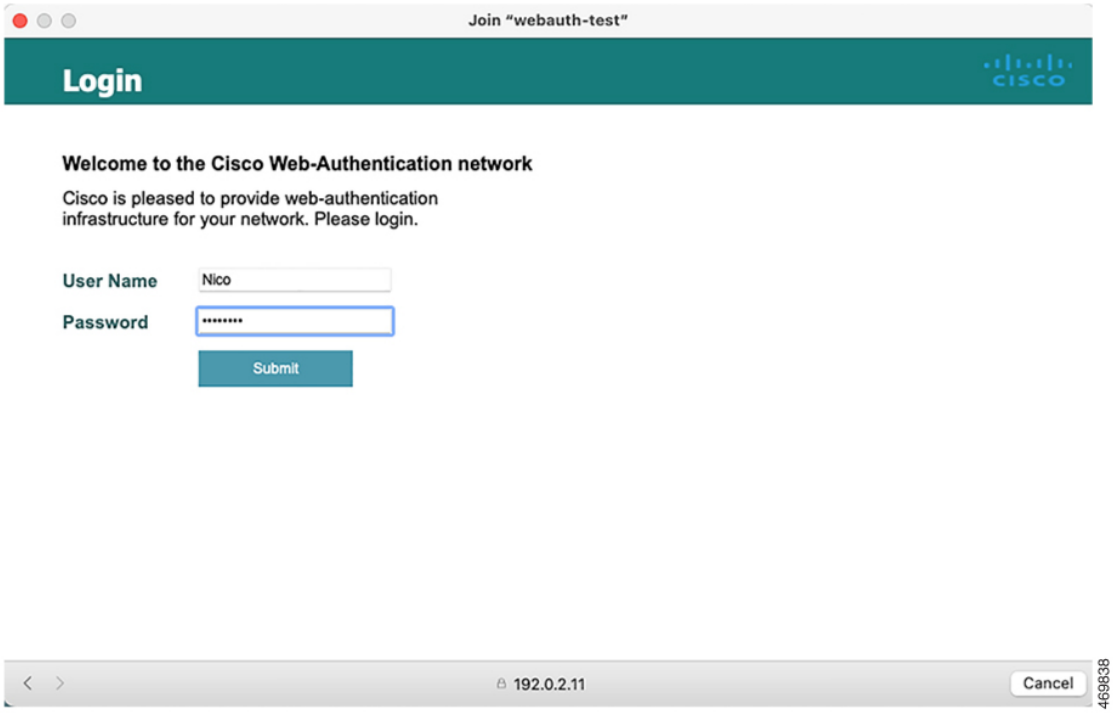
    banner file <filepath>

Figure 3: Customized Web Banner



If you do not enable a banner, only the username and password dialog boxes appear in the web authentication login screen, and no banner appears when you log into the switch.

Figure 4: Login Screen With No Banner



Customized Local Web Authentication

During the local web authentication process, the switch's internal HTTP server hosts four HTML pages to deliver to an authenticating client. The server uses these pages to notify you of these four authentication process states:

- Login: Your credentials are requested
- Success: The login was successful
- Fail: The login failed
- Expire: The login session has expired because of excessive login failures

Note Virtual IP address is mandatory to configure custom web authentication.

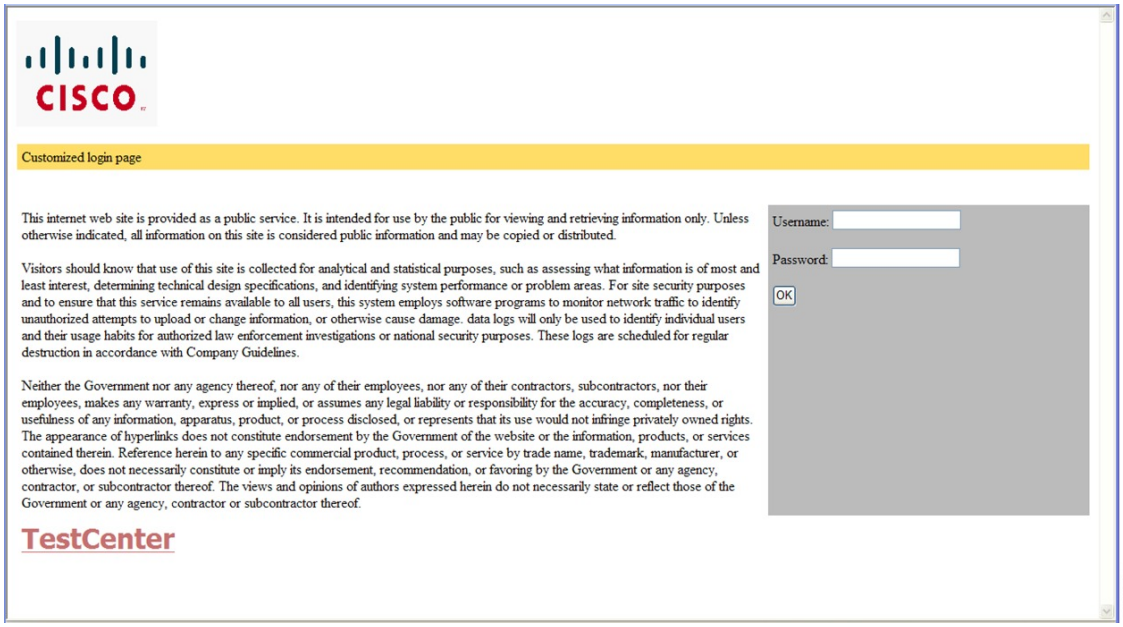
Guidelines

- You can substitute your own HTML pages for the default internal HTML pages.
- You can use a logo or specify text in the *login*, *success*, *failure*, and *expire* web pages.
- On the banner page, you can specify text in the login page.
- The pages are in HTML.

- You must include an HTML redirect command in the success page to access a specific URL.
- The URL string must be a valid URL (for example, http://www.cisco.com). An incomplete URL might cause *page not found* or similar errors on a web browser.
- If you configure web pages for HTTP authentication, they must include the appropriate HTML commands (for example, to set the page time out, to set a hidden password, or to confirm that the same page is not submitted twice). The custom page samples in the webauth bundle are provided with the image and the details of what you can and cannot change.
- The CLI command to redirect users to a specific URL is not available when the configured login form is enabled. The administrator should ensure that the redirection is configured in the web page.
- If the CLI command redirecting users to specific URL after authentication occurs is entered and then the command configuring web pages is entered, the CLI command redirecting users to a specific URL does not take effect.
- Configured web pages can be copied to the switch boot flash or flash.
- The login page can be on one flash, and the success and failure pages can be another flash (for example, the flash on the active switch or a member switch).
- You must configure all four pages.
- All of the logo files (image, flash, audio, video, and so on) that are stored in the system directory (for example, flash, disk0, or disk) and that are displayed on the login page must use *web_auth_<filename>* as the file name.
- The configured authentication proxy feature supports both HTTP and SSL.

You can substitute your HTML pages for the default internal HTML pages. You can also specify a URL to which users are redirected after authentication occurs, which replaces the internal Success page.

Figure 5: Customizable Authentication Page



Redirection URL for Successful Login Guidelines

When configuring a redirection URL for successful login, consider these guidelines:

- If the custom authentication proxy web pages feature is enabled, the redirection URL feature is disabled and is not available in the CLI. You can perform redirection in the custom-login success page.
- If the redirection URL feature is enabled, a configured auth-proxy-banner is not used
- To remove the specification of a redirection URL, use the **no** form of the command.
- If the redirection URL is required after the web-based authentication client is successfully authenticated, then the URL string must start with a valid URL (for example, http://) followed by the URL information. If only the URL is given without http://, then the redirection URL on successful authentication might cause page not found or similar errors on a web browser.

How to Configure Local Web Authentication

Configuring Default Local Web Authentication

The following table shows the default configurations required for local web authentication.

Table 1: Default Local Web Authentication Configuration

Feature	Default Setting
AAA	Disabled
RADIUS server • IP address • UDP authentication port • Key	• None specified
Default value of inactivity timeout	3600 seconds
Inactivity timeout	Disabled

Information About the AAA Wizard

The AAA wizard helps you to add the authentication, authorization, and accounting details without having to access multiple windows.

Note When command authorization is enabled as a part of AAA Authorization configuration through TACACS and the corresponding method list is not configured as a part of the HTTP configuration, WebUI pages will not load any data. However, some wireless feature pages may work as they are privilege-based and not command based.

Note Note the following limitations for a TACACS+ user on the 9800 WebUI:

- Users with privilege level 1-10 can only view the **Monitor** tab.
- Users with privilege level 15 have full access.
- Users with privilege level 15 and a command set allowing specific commands only, is not supported.

Note When you configure the AAA authentication and authorization attributes, the following format must be followed:

- protocol:attr=bla
- protocol:attr#0=bla
- protocol:attr#*=bla
- attr=bla
- attr#0=bla
- attr#*=bla

attr is mapped to the supported AAA attributes. If *attr* is an unknown or undefined attribute, a warning message *parse unknown cisco vsa* is displayed when you configure the **radius-server disallow unknown vendor-code** command. Otherwise, the transaction will be treated as a failure.

We recommend that you configure the command as per the format discussed above. Otherwise, the transaction fails. Whenever the passed attribute does not match any of the patterns mentioned, then AAA fails to decode that specific attribute and marks the request as a failure.

To edit the details entered using the wizard, use the respective screens.

Procedure

Step 1 Choose **Configuration** > **Security** > **AAA**.

Step 2 Click **+ AAA Wizard**.

The **Add Wizard** page is displayed.

Step 3 Click **RADIUS** tab.

The RADIUS server option is enabled by default. You can switch between the **Basic** and **Advanced** options using the radio buttons.

a) In the **Name** field, enter the name of the RADIUS server.
b) In the **IPv4 / IPv6 Server Address** field, enter the IPv4 or IPv6 address, or hostname.
c) Check the **PAC Key** check box to enable the Protected Access Credential (PAC) authentication key option.
d) From the **Key Type** drop-down list, choose the authentication key type.

e) In the **Key** field, enter the authentication key.
f) In the **Confirm Key** field, re-enter the authentication key.
g) Click the **Advanced** radio button.

This enables the **Advanced** options.

h) In the **Auth Port** field, enter the authorization port number.
i) In the **Acct Port** field, enter the accounting port number.
j) In the **Server Timeout** field, enter the timeout duration, in seconds.
k) In the **Retry Count** field, enter the number of retries.
l) Use the **Support for CoA** toggle button to enable or disable change of authorization (CoA).

Step 4 Check the **TACACS+** check box.

This enables the TACACS+ options. You can switch between the **Basic** and **Advanced** options using the radio buttons.

a) In the **Name** field, enter the TACACS+ server name.
b) In the **IPv4 / IPv6 Server Address** field, enter the IPv4 or IPv6 address, or hostname.
c) In the **Key** field, enter the authentication key.
d) In the **Confirm Key** field, re-enter the authentication key.
e) Click the **Advanced** radio button.

This enables the **Advanced** options.

f) In the **Port** field, enter the port number to use.
g) In the **Server Timeout** field, enter the timeout duration, in seconds.

Step 5 Check the **LDAP** check box.

This enables the LDAP options. You can switch between the **Basic** and **Advanced** options using the radio buttons.

a) In the Server Name field, enter the **LDAP** server name.
b) In the **IPv4 / IPv6 Server Address** field, enter the IPv4 or IPv6 address, or hostname.
c) In the **Port Number** field, enter the port number to use.
d) From the **Simple Bind** drop-down list, choose the authentication key type.
e) In the **User Base DN** field, enter the details.
f) Click the **Advanced** radio button.

This enables the **Advanced** options.

g) From the **User Attribute** drop-down list, choose the user attribute.
h) In the **User Object Type** field, enter the object type details and click the + icon.

The objects that have been added are listed in the area below. Use the x mark adjacent to each object to remove it.

i) In the **Server Timeout** field, enter the timeout duration, in seconds.
j) Check the **Secure Mode** check box to enable secure mode.

Checking this enables the **Trustpoint Name** drop-down list.

k) From the **Trustpoint Name** drop-down list, choose the trustpoint.
l) Click **Next**.

This enables the **Server Group Association** page and the RADIUS tab is selected by default.

Step 6 Perform the following actions under **RADIUS** tab.

a) In the **Name** field, enter the name of the RADIUS server group.
b) From the **MAC-Delimiter** drop-down list, choose the delimiter to be used in the MAC addresses that are sent to the RADIUS servers.
c) From the **MAC Filtering** drop-down list, choose a value based on which to filter MAC addresses.
d) To configure the dead time for the server group and direct AAA traffic to alternative groups of servers that have different operational characteristics, in the **Dead-Time** field, enter the amount of time, in minutes, after which a server is assumed to be dead.
e) Choose the servers that you want to include in the server group from the **Available Servers** list and move them to the **Assigned Servers** list.
f) Click **Next**.

The **TACACS+** window is displayed, if you have selected **TACACS+** in server configuration.

Step 7 Use the **TACACS+** window to enter the following details:

a) In the **Name** field, enter the name of the TACACS+ server group.
b) From the **Available Servers** list, choose the servers that you want to include in the server group from the list and move them to the **Assigned Servers** list.
c) Click **Next**.

The **LDAP** window is displayed, if you have selected **LDAP** under server configuration.

Step 8 Use the **LDAP** window to enter the following details:

a) In the **Name** field, enter the name of the LDAP server group.
b) From the **Available Servers** list, choose the servers that you want to include in the server group from the list and move them to the **Assigned Servers** list.

Step 9 Click **Next**.

The **MAP AAA** window is displayed.

Use the check boxes to enable the **Authentication**, **Authorization**, and **Accounting** tabs. You cannot unselect all the three options. At least one option has to be selected.

Step 10 Use the **Authentication** tab to enter the authentication details:

a) In the **Method List Name** field, enter the name of the method list.
b) From the **Type** drop-down list, choose the type of accounting that you want to perform before allowing access to the network.
c) From the **Group Type** drop-down list, choose a value depending on whether you want to assign a group of servers as your access server, or want to use a local server to authenticate access.

If you choose the local option, the **Fallback** to local option is removed.

d) Check the **Fallback to local** check box to configure a local server to act as a fallback method when servers in the group are unavailable.
e) From the **Available Server Groups** list, choose the server groups that you want to use to authenticate access to your network and click the > icon to move them to the **Assigned Server Groups** list.

Step 11 Check the **Authorization** check box to configure the authorization details:

a) In the **Method List Name** field, enter the name of the method list.
b) From the **Type** drop-down list, choose the type of authorization you want to perform before allowing access to the network.

c) From the **Group Type** drop-down list, choose a value depending on whether you want to assign a group of servers as your access server, or want to use a local server to authorize access.

If you choose the local option, the **Fallback** to local option is removed.

d) Check the **Fallback to local** check box to configure a local server to act as a fallback method when the servers in the group are unavailable.

e) From the **Available Server Groups** list, choose the server groups you want to use to authorize access to your network and click > icon to move them to the **Assigned Server Groups** list.

Step 12 Check the **Accounting** check box to configure the accounting details:

a) In the **Method List Name** field, enter the name of the method list.

b) From the **Type** drop-down list, choose the type of accounting that you want to perform.

c) From the **Available Server Groups** list, choose the server groups that you want to use to authorize access to your network and click the > icon to move them to the **Assigned Server Groups** list.

Step 13 Click **Apply to Device**.

Configuring AAA Authentication (GUI)

Note The WebUI does not support the ipv6 radius source-interface under AAA radius server group configuration.

Procedure

Step 1 Choose **Configuration** > **Security** > **AAA**.

Step 2 In the **Authentication** section, click **Add**.

Step 3 In the **Quick Setup: AAA Authentication** window that is displayed, enter a name for your method list.

Step 4 Choose the type of authentication you want to perform before allowing access to the network, in the **Type** drop-down list.

Step 5 Choose if you want to assign a group of servers as your access server, or if you want to use a local server to authenticate access, from the **Group** Type drop-down list.

Step 6 To configure a local server to act as a fallback method when servers in the group are unavailable, check the **Fallback** to local check box.

Step 7 Choose the server groups you want to use to authenticate access to your network, from the **Available Server Groups** list and click > icon to move them to the **Assigned Server Groups** list.

Step 8 Click **Save & Apply to Device**.

Configuring AAA Authentication (CLI)

Procedure

	Command or Action	Purpose
Step 1	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables AAA functionality.
Step 2	**aaa authentication login** {**default** | *named_authentication_list*} **group** *AAA_group_name* **Example:** `Device(config)# aaa authentication login default group group1`	Defines the list of authentication methods at login. **named_authentication_list** refers to any name that is not greater than 31 characters. **AAA_group_name** refers to the server group name. You need to define the server-group **server_name** at the beginning itself.
Step 3	**aaa authorization network** {**default** | **named**} **group** *AAA_group_name* **Example:** `Device(config)# aaa authorization network default group group1`	Creates an authorization method list for web-based authorization.
Step 4	**tacacs server** *server-name* **Example:** `Device(config)# tacacs server yourserver`	Specifies an AAA server.
Step 5	**address** {**ipv4** | **ipv6**}*ip_address* **Example:** `Device(config-server-tacacs)# address ipv4 10.0.1.12`	Configures the IP address for the TACACS server.
Step 6	**single-connection** **Example:** `Device(config-server-tacacs)# single-connection`	Multiplexes all packets over a single TCP connection to TACACS server.
Step 7	**tacacs-server host** {*hostname* | *ip_address*} **Example:**	Specifies a AAA server.

	Command or Action	Purpose
	`Device(config)# tacacs-server host 10.1.1.1`	

Configuring the HTTP/HTTPS Server (GUI)

Procedure

Step 1 Choose **Administration** > **Management** > **HTTP/HTTPS/Netconf**.

Step 2 In the **HTTP/HTTPS Access Configuration** section, enable HTTP Access and enter the port that will listen for HTTP requests. The default port is 80. Valid values are 80, and ports between 1025 and 65535.

Step 3 Enable **HTTPS Access** on the device and enter the designated port to listen for HTTPS requests. The default port is 1025. Valid values are 443, and ports between 1025 and 65535. On a secure HTTP connection, data to and from an HTTP server is encrypted before being sent over the Internet. HTTP with SSL encryption provides a secure connection to allow such functions as configuring a switch from a Web browser.

Step 4 Choose the **Personal Identity Verification** as enabled or disabled.

Step 5 In the **HTTP Trust Point Configuration** section, enable **Enable Trust Point** to use Certificate Authority servers as trustpoints.

Step 6 From the **Trust Points** drop-down list, choose a trust point.

Step 7 In the **Timeout Policy Configuration** section, enter the HTTP timeout policy in seconds. Valid values can range from 1 to 600 seconds.

Step 8 Enter the number of minutes of inactivity allowed before the session times out. Valid values can range from 180 to 1200 seconds.

Step 9 Enter the server life time in seconds. Valid values can range from 1 to 86400 seconds.

Step 10 Enter the maximum number of requests the device can accept. Valid values range from 1 to 86400 requests.

Step 11 Save the configuration.

Configuring the HTTP Server (CLI)

To use local web authentication, you must enable the HTTP server within the device. You can enable the server for either HTTP or HTTPS.

Note The Apple psuedo-browser will not open if you configure only the **ip http secure-server** command. You should also configure the **ip http server** command.

Follow the procedure given below to enable the server for either HTTP or HTTPS:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**ip http server** **Example:** `Device(config)# ip http server`	Enables the HTTP server. The local web authentication feature uses the HTTP server to communicate with the hosts for user authentication.
Step 3	**ip http secure-server** **Example:** `Device(config)# ip http secure-server`	Enables HTTPS. You can configure custom authentication proxy web pages or specify a redirection URL for successful login. **Note** To ensure secure authentication when you enter the **ip http secure-server** command, the login page is always in HTTPS (secure HTTP) even if the user sends an HTTP request.
Step 4	**end** **Example:** `Device(config)# end`	Exits configuration mode.

Configuring HTTP and HTTPS Requests for Web Authentication

Information About Configuring HTTP and HTTPS Requests for Web Authentication

Using the Configuring HTTP and HTTPS Requests for Web Authentication feature, you can have HTTPS access to device management and HTTP access to web authentication. To control the HTTP and HTTPS requests being sent to the web authentication module, run the **secure-webauth-disable** and **webauth-http-enable** commands in the global parameter map mode.

Note The **secure-webauth-disable** and **webauth-http-enable** commands are not enabled by default; you must configure them explicitly.

The following table describes the various CLI combinations:

Table 2: CLI Combinations

Admin (Device Management)		WebAuthentication		Required Configurations	
HTTP Access	**HTTPS Access**	**HTTP Access**	**HTTPS Access**	**Admin**	**Web Authentication**

No	Yes	Yes	Yes	`no ip http server`
				
				`ip http secure-server`
					`no ip http server`
					`ip http secure-server`
					
					`parameter-map type webauth global`
					
					`webauth-http-enable`

No	Yes	No	Yes	`no ip http server`
				
				`ip http secure-server`
					`no ip http server`
					`ip http secure-server`

No	Yes	Yes	No	`no ip http server`
				`ip http secure-server`
					`no ip http server`
					`ip http secure-server`
					
					`parameter-map type webauth global`
					
					`webauth-http-enable`
					
					`secure-webauth-disable`

No	Yes	No	No	`no ip http server`
				`ip http secure-server`
					`no ip http server`
					`ip http secure-server`
					
					`parameter-map type webauth global`
					
					`secure-webauth-disable`

No	No	No	Yes	`no ip http server` `no ip http secure-server`	Not Supported

No	No	Yes	No	`no ip http server`
				`no ip http secure-server`
					`no ip http server`
					`no ip http secure-server`
					
					`parameter-map type webauth global`
					
					`webauth-http-enable`

Yes	No	Yes	No	`ip http server` `no ip http secure-server`	`ip http server` `no ip http secure-server`

Admin (Device Management)		WebAuthentication		Required Configurations	
HTTP Access	HTTPS Access	HTTP Access	HTTPS Access	Admin	Web Authentication

Yes	Yes	Yes	No	`ip http server`
				`ip http secure-server`
					`ip http server`
					`ip http secure-server`
					
					`parameter-map type webauth global`
					
					`secure-webauth-disable`

Note

- The **ip http server** and **ip http secure-server** commands allow access for HTTP and HTTPS, respectively. For example, in the first row of the table, for HTTP access to web authentication, you do not require the **ip http server** command. You can use the new **webauth-http-enable** command under the global parameter map, to allow HTTP access.
- For HTTPS access to webauth, the **ip http secure-server** command is required. Therefore, HTTPS access for both admin and web authentication are enabled in the first row. To disable HTTPS access for web authentication, configure the **secure-webauth-disable** command. For example, in the fourth row of the table, HTTPS access is disabled for web authentication because the **secure-webauth-disable** command is configured.

Guidelines and Limitations

The following are the guidelines and limitations for configuring HTTP and HTTPS requests for web authentication:

- You cannot enable HTTPS web authentication without enabling HTTPS for device management.
- If the **secure-webauth-disable** command is configured, central web authentication cannot be performed, if the initial request from the client is https://< >.

Configuring HTTP and HTTPS Requests for Web Authentication (CLI)

To configure the HTTP and HTTPS requests being sent to the webauth module, complete the steps given below:

Procedure

	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	**no ip http server** **Example:** `Device(config)# no ip http server`	Sets the HTTP server to its default.
Step 4	**ip http** {**server** | **secure-server**} **Example:** `Device(config)# ip http server`	Enables the HTTP server or the HTTP secure server.
Step 5	**parameter-map type webauth global** **Example:** `Device(config)# parameter-map type webauth global`	Enables the global parameter map mode.
Step 6	**secure-webauth-disable** **Example:** `Device(config-params-parameter-map)# secure-webauth-disable`	Disables HTTP secure server for web authentication.
Step 7	**webauth-http-enable** **Example:** `Device(config-params-parameter-map)# webauth-http-enable`	Enables HTTP server for web authentication.

Creating a Parameter Map (GUI)

Procedure

Step 1 Choose **Configuration** > **Security** > **Web Auth**.

Step 2 Click **Add**.

Step 3 Click **Policy Map**.

Step 4 Enter **Parameter Name**, **Maximum HTTP connections**, **Init-State Timeout(secs)** and choose **webauth** in the **Type** drop-down list.

Step 5 Click **Apply to Device**.

Creating Parameter Maps

Configuring Local Web Authentication (GUI)

Procedure

Step 1 Choose **Configuration** > **Security** > **Web Auth**.

Step 2 On the **Web Auth** page, click **Add**.

Step 3 In the **Create Web Auth Parameter** window that is displayed, enter a name for the parameter map.

Step 4 In the **Maximum HTTP Connections** field, enter the maximum number of HTTP connections that you want to allow.

Step 5 In the **Init-State Timeout** field, enter the time after which the init state timer should expire due to user's failure to enter valid credentials in the login page.

Step 6 Choose the type of Web Auth parameter.

Step 7 Click **Apply to Device**.

Step 8 On the **Web Auth** page, click the name of the parameter map.

Step 9 In the **Edit WebAuth Parameter** window that is displayed, choose the required **Banner Type**.

- If you choose **Banner Text**, enter the required banner text to be displayed.
- If you choose **File Name**, specify the path of the file from which the banner text has to be picked up.

Step 10 Enter the virtual IP addresses as required.

Step 11 Set appropriate status of **WebAuth Intercept HTTPS**, **Captive Bypass Portal**.

Step 12 Set appropriate status for **Disable Success Window**, **Disable Logout Window**, and **Login Auth Bypass for FQDN**.

Step 13 Check the **Sleeping Client Status** check box to enable authentication of sleeping clients and then specify the **Sleeping Client Timeout** in minutes. Valid range is between 10 minutes and 43200 minutes.

Step 14 Click the **Advanced** tab.

Step 15 To configure external web authentication, perform these tasks:

a) In the **Redirect for log-in** field, enter the name of the external server to send login request.
b) In the **Redirect On-Success** field, enter the name of the external server to redirect after a successful login.
c) In the **Redirect On-Failure** field, enter the name of the external server to redirect after a login failure.
d) (Optional) Under **Redirect to External Server** in the **Redirect Append for AP MAC Address** field, enter the AP MAC address.
e) (Optional) In the **Redirect Append for Client MAC Address** field, enter the client MAC address.
f) (Optional) In the **Redirect Append for WLAN SSID** field, enter the WLAN SSID.
g) In the **Portal IPV4 Address** field, enter the IPv4 address of the portal to send redirects.
h) In the **Portal IPV6 Address** field, enter the IPv6 address of the portal to send redirects, if IPv6 address is used.

Step 16 To configure customized local web authentication, perform these tasks:

a) Under **Customized Page**, specify the following pages:

- **Login Failed Page**
- **Login Page**

- **Logout Page**
- **Login Successful Page**

Step 17 Click **Update & Apply**.

Configuring the Internal Local Web Authentication (CLI)

Follow the procedure given below to configure the internal local web authentication:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**parameter-map type webauth** {*parameter-map-name* | **global**} **Example:** `Device(config)# parameter-map type webauth sample`	Creates the parameter map. The **parameter-map-name** must not exceed 99 characters.
Step 3	**end** **Example:** `Device(config-params-parameter-map)# end`	Returns to privileged EXEC mode.

Configuring the Customized Local Web Authentication (CLI)

Follow the procedure given below to configure the customized local web authentication:

Note Virtual IP address is mandatory for custom web authentication.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 2	**parameter-map type webauth** *parameter-map-name* **Example:** `Device(config)# parameter-map type webauth sample`	Configures the webauth type parameter. **Note** You need to configure a virtual IP in the global parameter map to use the customized web authentication bundle.
Step 3	**type** {**authbypass** | **consent** | **webauth** | **webconsent**} **Example:** `Device(config-params-parameter-map)# type webauth`	Configures webauth sub-types, such as passthru, consent, webauth, or webconsent.
Step 4	**custom-page login device** *html-filename* **Example:** `Device(config-params-parameter-map)# custom-page login device bootflash:login.html`	Configures the customized login page.
Step 5	**custom-page login expired device** *html-filename* **Example:** `Device(config-params-parameter-map)# custom-page login expired device bootflash:loginexpired.html`	Configures the customized login expiry page.
Step 6	**custom-page success device** *html-filename* **Example:** `Device(config-params-parameter-map)# custom-page success device bootflash:loginsuccess.html`	Configures the customized login success page.
Step 7	**custom-page failure device** *html-filename* **Example:** `Device(config-params-parameter-map)# custom-page failure device bootflash:loginfail.html`	Configures the customized login failure page.
Step 8	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-params-parameter-map)# end`	

Configuring the External Local Web Authentication (CLI)

Follow the procedure given below to configure the external local web authentication:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**parameter-map type webauth** *parameter-map-name* **Example:** `Device(config)# parameter-map type webauth sample`	Configures the webauth type parameter.
Step 3	**type** {**authbypass** | **consent** | **webauth** | **webconsent**} **Example:** `Device(config-params-parameter-map)# type webauth`	Configures the webauth sub-types, such as authbypass, consent, passthru, webauth, or webconsent.
Step 4	**redirect** [**for-login** | **on-failure** | **on-success**] *URL* **Example:** `Device(config-params-parameter-map)# redirect for-login http://www.cisco.com/login.html`	Configures the redirect URL for the login, failure, and success pages. **Note** In the redirect url, you need to press *Ctrl+v* and type *?* to configure the *?* character. The *?* character is commonly used in URL when ISE is configured as an external portal.
Step 5	**redirect portal** {**ipv4** | **ipv6**} *ip-address* **Example:**	Configures the external portal IPv4 address.

	Command or Action	Purpose
	`Device(config-params-parameter-map)# redirect portal ipv4 23.0.0.1`	**Note** The IP address should be one of the associated IP addresses of the domain and not a random IP address when using FQDN. It is recommended to use the FQDN URL here, if a given domain resolves to more than a single IP address.
Step 6	**end** **Example:** `Device(config-params-parameter-map)# end`	Returns to privileged EXEC mode.

Configuring the Web Authentication WLANs

Follow the procedure given below to configure WLAN using web auth security and map the authentication list and parameter map:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**wlan** *profile-name wlan-id ssid-name* **Example:** `Device(config)# wlan mywlan 34 mywlan-ssid`	Specifies the WLAN name and ID. *profile-name* is the WLAN name which can contain 32 alphanumeric characters. *wlan-id* is the wireless LAN identifier. The valid range is from 1 to 512. *ssid-name* is the SSID which can contain 32 alphanumeric characters.
Step 3	**no security wpa** **Example:** `Device(config-wlan)# no security wpa`	Disables the WPA security.
Step 4	**security web-auth** **Example:** `Device(config-wlan)# security web-auth`	Enables web authentication for WLAN.

	Command or Action	Purpose

Step 5	**security web-auth** {**authentication-list** *authentication-list-name* | **parameter-map** *parameter-map-name*}
	Example:
	`Device(config-wlan)# security web-auth authentication-list webauthlistlocal`
	`Device(config-wlan)# security web-auth parameter-map sample`
		Enables web authentication for WLAN.
		Here,
		• **authentication-list** *authentication-list-name*: Sets the authentication list for IEEE 802.1x.
		• **parameter-map** *parameter-map-name*: Configures the parameter map.
		Note When **security web-auth** is enabled, you get to map the default **authentication-list** and global **parameter-map**. This is applicable for authentication-list and parameter-map that are not explicitly mentioned.

Step 6	**end** **Example:** `Device(config-wlan)# end`	Returns to privileged EXEC mode.

Configuring Pre-Auth Web Authentication ACL (GUI)

Before you begin

Ensure that you have configured an access control list (ACL) and a WLAN.

Procedure

Step 1 Choose **Configuration** > **Tags & Profiles** > **WLANs**.
Step 2 Click the name of the WLAN.
Step 3 In the **Edit WLAN** window, click the **Security** tab and then click the **Layer3** tab.
Step 4 Click **Show Advanced Settings**.
Step 5 In the **Preauthenticaion ACL** section, choose the appropriate ACL to be mapped to the WLAN.
Step 6 Click **Update & Apply to Device**.

Configuring Pre-Auth Web Authentication ACL (CLI)

Follow the procedure given below to configure pre-auth web authentication ACL:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**access-list** *access-list-number* {**deny** | **permit**} *hostname source-wildcard-bits* **Example:** `Device(config)# access-list 2 deny your_host 10.1.1.1 log`	Creates an ACL list. The *access-list-number* is a decimal number from 1 to 99, 100 to 199, 300 to 399, 600 to 699, 1300 to 1999, 2000 to 2699, or 2700 to 2799. Enter **deny** or **permit** to specify whether to deny or permit if the conditions are matched. The *source* is the source address of the network or host from which the packet is being sent specified as: • The 32-bit quantity in dotted-decimal format. • The keyword **any** as an abbreviation for *source* and *source-wildcard* of 0.0.0.0 255.255.255.255. You do not need to enter a source-wildcard. • The keyword **host** as an abbreviation for *source* and *source-wildcard* of source 0.0.0.0. (Optional) The *source-wildcard* applies wildcard bits to the source.
Step 3	**wlan** *profile-name wlan-id ssid-name* **Example:** `Device(config)# wlan mywlan 34 mywlan-ssid`	Creates the WLAN. *profile-name* is the WLAN name which can contain 32 alphanumeric characters. *wlan-id* is the wireless LAN identifier. The valid range is from 1 to 512. *ssid-name* is the SSID which can contain 32 alphanumeric characters.
Step 4	**ip access-group web** *access-list-name* **Example:** `Device(config-wlan)# ip access-group web name`	Maps the ACL to the web auth WLAN. *access-list-name* is the IPv4 ACL name or ID.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-wlan)# end`	Returns to privileged EXEC mode.

Configuring the Maximum Web Authentication Request Retries

Follow these steps to configure the maximum web authentication request retries:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**wireless security web-auth retries** *number* **Example:** `Device(config)# wireless security web-auth retries 2`	*number* is the maximum number of web auth request retries. The valid range is 0 to 20.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring a Local Banner in Web Authentication Page (GUI)

Procedure

Step 1 Choose **Configuration** > **Security** > **Web Auth**.

Step 2 In the **Webauth Parameter Map** tab, click the parameter map name. The **Edit WebAuth Parameter** window is displayed.

Step 3 In the **General** tab and choose the required Banner Type:

- If you choose **Banner Text**, enter the required banner text to be displayed.
- If you choose **File Name**, specify the path of the file from which the banner text has to be picked up.

Step 4 Click **Update & Apply**.

Configuring a Local Banner in Web Authentication Page (CLI)

Follow the procedure given below to configure a local banner in web authentication pages.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**parameter-map type webauth** *param-map* **Example:** `Device(config)# parameter-map type webauth param-map`	Configures the web authentication parameters. Enters the parameter map configuration mode.
Step 3	**banner** [*file* | *banner-text* | *title*] **Example:** `Device(config-params-parameter-map)# banner http C My Switch C`	Enables the local banner. Create a custom banner by entering *C banner-text C* (where *C* is a delimiting character), or *file* that indicates a file (for example, a logo or text file) that appears in the banner, or *title* that indicates the title of the banner.
Step 4	**end** **Example:** `Device(config-params-parameter-map)# end`	Returns to privileged EXEC mode.

Configuring Type WebAuth, Consent, or Both

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device # configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**parameter-map type webauth** *parameter-map name* **Example:** `Device (config) # parameter-map type webauth webparalocal`	Configures the webauth type parameter.
Step 3	**type consent** **Example:** `Device (config-params-parameter-map) # type consent`	Configures webauth type as consent. You can configure the type as webauth, consent, or both (webconsent).
Step 4	**end** **Example:** `Device (config-params-parameter-map) # end`	Returns to privileged EXEC mode.
Step 5	**show running-config | section parameter-map type webauth** *parameter-map* **Example:** `Device (config) # show running-config | section parameter-map type webauth test`	Displays the configuration details.

Configuring Preauthentication ACL

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**wlan** *wlan-name* **Example:** `Device (config)# wlan ramban`	For *wlan-name*, enter the profile name.
Step 3	**shutdown** **Example:** `Device (config-wlan)# shutdown`	Disables the WLAN.
Step 4	**ip access-group web** *preauthrule* **Example:** `Device (config-wlan)# ip access-group web preauthrule`	Configures ACL that has to be applied before authentication.

	Command or Action	Purpose
Step 5	**no shutdown** **Example:** `Device (config)# no shutdown`	Enables the WLAN.
Step 6	**end** **Example:** `Device (config-wlan)# end`	Returns to privileged EXEC mode.
Step 7	**show wlan name** *wlan-name* **Example:** `Device# show wlan name ramban`	Displays the configuration details.

Configuring TrustPoint for Local Web Authentication

Before you begin

Ensure that a certificate is installed on your controller . Using trustpoint controller presents the domain specific certificate that client browser trusts when it gets redirected to ***.com** portal.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**parameter-map type webauth global** **Example:** `Device (config)# parameter-map type webauth global`	Creates the parameter map.
Step 3	**trustpoint** *trustpoint-name* **Example:** `Device (config-params-parameter-map)# trustpoint trustpoint-name`	Configures trustpoint for local web authentication.
Step 4	**end** **Example:** `Device (config-params-parameter-map)# end`	Returns to privileged EXEC mode.

Configuration Examples for Local Web Authentication

Example: Obtaining Web Authentication Certificate

This example shows how to obtain web authentication certificate.

```
Device# configure terminal
Device(config)# crypto pki import cert pkcs12 tftp://9.1.0.100/ldapserver-cert.p12 cisco
Device(config)# end
Device# show crypto pki trustpoints cert
 Trustpoint cert:
    Subject Name:
    e=rkannajr@cisco.com
    cn=sthaliya-lnx
    ou=WNBU
    o=Cisco
    l=SanJose
    st=California
    c=US
          Serial Number (hex): 00
    Certificate configured.
Device# show  crypto pki certificates cert
Certificate
  Status: Available
  Certificate Serial Number (hex): 04
  Certificate Usage: General Purpose
  Issuer:
    e=rkannajr@cisco.com
    cn=sthaliya-lnx
    ou=WNBU
    o=Cisco
    l=SanJose
    st=California
    c=US
  Subject:
    Name: ldapserver
    e=rkannajr@cisco.com
    cn=ldapserver
    ou=WNBU
    o=Cisco
    st=California
    c=US
  Validity Date:
    start date: 07:35:23 UTC Jan 31 2012
    end   date: 07:35:23 UTC Jan 28 2022
  Associated Trustpoints: cert ldap12
  Storage: nvram:rkannajrcisc#4.cer

CA Certificate
  Status: Available
  Certificate Serial Number (hex): 00
  Certificate Usage: General Purpose
  Issuer:
    e=rkannajr@cisco.com
    cn=sthaliya-lnx
    ou=WNBU
    o=Cisco
    l=SanJose
    st=California
```

```
    c=US
  Subject:
    e=rkannajr@cisco.com
    cn=sthaliya-lnx
    ou=WNBU
    o=Cisco
    l=SanJose
    st=California
    c=US
  Validity Date:
    start date: 07:27:56 UTC Jan 31 2012
    end   date: 07:27:56 UTC Jan 28 2022
  Associated Trustpoints: cert ldap12 ldap
  Storage: nvram:rkannajrcisc#0CA.cer
```

Example: Displaying a Web Authentication Certificate

This example shows how to display a web authentication certificate.

```
Device# show crypto ca certificate verb
     Certificate
     Status: Available
     Version: 3
     Certificate Serial Number (hex): 2A9636AC00000000858B
     Certificate Usage: General Purpose
     Issuer:
    cn=Cisco Manufacturing CA
    o=Cisco Systems
    Subject:
    Name: WS-C3780-6DS-S-2037064C0E80
    Serial Number: PID:WS-C3780-6DS-S SN:FOC1534X12Q
    cn=WS-C3780-6DS-S-2037064C0E80
    serialNumber=PID:WS-C3780-6DS-S SN:FOC1534X12Q
    CRL Distribution Points:
    http://www.cisco.com/security/pki/crl/cmca.crl
    Validity Date:
    start date: 15:43:22 UTC Aug 21 2011
    end   date: 15:53:22 UTC Aug 21 2021
    Subject Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (1024 bit)
    Signature Algorithm: SHA1 with RSA Encryption
    Fingerprint MD5: A310B856 A41565F1 1D9410B5 7284CB21
    Fingerprint SHA1: 04F180F6 CA1A67AF 9D7F561A 2BB397A1 0F5EB3C9
    X509v3 extensions:
    X509v3 Key Usage: F0000000
      Digital Signature
      Non Repudiation
      Key Encipherment
      Data Encipherment
    X509v3 Subject Key ID: B9EEB123 5A3764B4 5E9C54A7 46E6EECA 02D283F7
    X509v3 Authority Key ID: D0C52226 AB4F4660 ECAE0591 C7DC5AD1 B047F76C
    Authority Info Access:
    Associated Trustpoints: CISCO_IDEVID_SUDI
    Key Label: CISCO_IDEVID_SUDI
```

Example: Choosing the Default Web Authentication Login Page

This example shows how to choose a default web authentication login page.

```
Device# configure terminal
Device(config)# parameter-map type webauth test
This operation will permanently convert all relevant authentication commands to their CPL
control-policy equivalents. As this conversion is irreversible and will
disable the conversion CLI 'authentication display [legacy|new-style]', you are strongly
advised to back up your current configuration before proceeding.
Do you wish to continue? [yes]: yes
Device(config)# wlan wlan50
Device(config-wlan)# shutdown
Device(config-wlan)# security web-auth authentication-list test
Device(config-wlan)# security web-auth parameter-map test
Device(config-wlan)# no shutdown
Device(config-wlan)# end
Device# show running-config | section wlan50
wlan wlan50 50 wlan50
 security wpa akm cckm
 security wpa wpa1
 security wpa wpa1 ciphers aes
 security wpa wpa1 ciphers tkip
 security web-auth authentication-list test
 security web-auth parameter-map test
 session-timeout 1800
 no shutdown

Device# show running-config | section parameter-map type webauth test
parameter-map type webauth test
 type webauth
```

Example: Choosing a Customized Web Authentication Login Page from an IPv4 External Web Server

This example shows how to choose a customized web authentication login page from an IPv4 external web server.

```
Device# configure terminal
Device(config)# parameter-map type webauth global
Device(config-params-parameter-map)# virtual-ip ipv4 192.0.2.1.
Device(config-params-parameter-map)# parameter-map type webauth test
Device(config-params-parameter-map)# type webauth
Device(config-params-parameter-map)# redirect for-login http://9.1.0.100/login.html
Device(config-params-parameter-map)# redirect portal ipv4 9.1.0.100
Device(config-params-parameter-map)# end
Device# show running-config | section parameter-map
parameter-map type webauth global
virtual-ip ipv4 192.0.2.1.
parameter-map type webauth test
type webauth
redirect for-login http://9.1.0.100/login.html
redirect portal ipv4 9.1.0.100
security web-auth parameter-map rasagna-auth-map
security web-auth parameter-map test
```

Example: Choosing a Customized Web Authentication Login Page from an IPv6 External Web Server

This example shows how to choose a customized web authentication login page from an IPv6 external web server.

```
Device# configure terminal
Device(config)# parameter-map type webauth global
Device(config-params-parameter-map)# virtual-ip ipv6 2001:DB8::/48
Device(config-params-parameter-map)# parameter-map type webauth test
Device(config-params-parameter-map)# type webauth
Device(config-params-parameter-map)# redirect for-login http://9:1:1::100/login.html
Device(config-params-parameter-map)# redirect portal ipv6 9:1:1::100
Device(config-params-parameter-map)# end
Device# show running-config | section parameter-map
parameter-map type webauth global
virtual-ip ipv6 2001:DB8::/48
parameter-map type webauth test
type webauth
redirect for-login http://9:1:1::100/login.html
redirect portal ipv6 9:1:1::100
security web-auth parameter-map rasagna-auth-map
security web-auth parameter-map test
```

Example: Assigning Login, Login Failure, and Logout Pages per WLAN

This example shows how to assign login, login failure and logout pages per WLAN.

```
Device# configure terminal
Device(config)# parameter-map type webauth test
Device(config-params-parameter-map)# custom-page login device flash:loginsantosh.html
Device(config-params-parameter-map)# custom-page login expired device flash:loginexpire.html
Device(config-params-parameter-map)# custom-page failure device flash:loginfail.html
Device(config-params-parameter-map)# custom-page success device flash:loginsucess.html
Device(config-params-parameter-map)# end
Device# show running-config | section parameter-map type webauth test
 parameter-map type webauth test
 type webauth
 redirect for-login http://9.1.0.100/login.html
 redirect portal ipv4 9.1.0.100
 custom-page login device flash:loginsantosh.html
 custom-page success device flash:loginsucess.html
 custom-page failure device flash:loginfail.html
 custom-page login expired device flash:loginexpire.html
```

Example: Configuring Preauthentication ACL

This example shows how to configure preauthentication ACL.

```
Device# configure terminal
Device(config)# wlan fff
Device(config-wlan)# shutdown
Device(config-wlan)# ip access-group web preauthrule
Device(config-wlan)# no shutdown
```

```
Device(config-wlan)# end
Device# show wlan name fff
```

Example: Configuring Webpassthrough

This example shows how to configure webpassthrough.

```
Device# configure terminal
Device(config)# parameter-map type webauth webparalocal
Device(config-params-parameter-map)# type consent
Device(config-params-parameter-map)# end
Device# show running-config | section parameter-map type webauth test
 parameter-map type webauth test
 type webauth
 redirect for-login http://9.1.0.100/login.html
 redirect portal ipv4 9.1.0.100
```

Verifying Web Authentication Type

To verify the web authentication type, run the following command:

```
Device# show parameter-map type webauth all
Type Name
---------------------------------
Global global
Named webauth
Named ext
Named redirect
Named abc
Named glbal
Named ewa-2
```

```
Device# show parameter-map type webauth global
Parameter Map Name : global
Banner:
Text : CisCo
Type : webauth
Auth-proxy Init State time : 120 sec
Webauth max-http connection : 100
Webauth logout-window : Enabled
Webauth success-window : Enabled
Consent Email : Disabled
Sleeping-Client : Enabled
Sleeping-Client timeout : 60 min
Virtual-ipv4 : 192.0.2.1.
Virtual-ipv4 hostname :
Webauth intercept https : Disabled
Webauth Captive Bypass : Disabled
Webauth bypass intercept ACL :
Trustpoint name :
HTTP Port : 80
Watch-list:
Enabled : no
Webauth login-auth-bypass:
```

```
Device# show parameter-map type webauth name global
Parameter Map Name : global
Type : webauth
Auth-proxy Init State time : 120 sec
Webauth max-http connection : 100
Webauth logout-window : Enabled
```

```
Webauth success-window : Enabled
Consent Email : Disabled
Sleeping-Client : Disabled
Webauth login-auth-bypass:
```

External Web Authentication (EWA)

Configuring EWA with Single WebAuth Server Address and Default Ports (80/443) (CLI)

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**aaa authentication login** **Example:** `Device(config)# aaa authentication login WEBAUTH local`	Defines the authentication method at login.
Step 3	**parameter-map type webauth** *parameter-map-name* **Example:** `Device(config)# parameter-map type webauth ISE-Ext-Webauth_IP`	Creates the parameter map. The *parameter-map-name* must not exceed 99 characters.
Step 4	**type webauth** **Example:** `Device(config-params-parameter-map)# type webauth`	Configures the webauth type parameter.
Step 5	**redirect for-login** *URL-String* **Example:** `Device(config-params-parameter-map)# redirect for-login https://192.168.0.98:8443/portal/PortalSetup.action?portal=a6406210-0981-11e7-8591-0050568915b2`	Configures the URL string for redirect during login.
Step 6	**redirect portal ipv4** *ip-address* **Example:** `Device(config-params-parameter-map)# redirect portal ipv4 192.168.0.98`	Configures the external portal IPv4 address.
Step 7	**exit** **Example:**	Returns to global configuration mode.

	Command or Action	Purpose
	`Device(config-params-parameter-map)# exit`	
Step 8	**wlan** *wlan-name wlan-id SSID-name* **Example:** `Device(config)#  wlan EWLC3-GUEST 3 EWLC3-GUEST`	Configures a WLAN.
Step 9	**no security ft adaptive** **Example:** `Device(config-wlan)# no security ft adaptive`	Disables adaptive 11r.
Step 10	**no security wpa** **Example:** `Device(config-wlan)# no security wpa`	Disables WPA security.
Step 11	**no security wpa wpa2** **Example:** `Device(config-wlan)# no security wpa wpa2`	Disables WPA2 security.
Step 12	**no security wpa wpa2 ciphers aes** **Example:** `Device(config-wlan)# no security wpa wpa2 ciphers aes`	Disables WPA2 ciphers for AES.
Step 13	**no security wpa akm dot1x** **Example:** `Device(config-wlan)# no security wpa akm dot1x`	Disables security AKM for dot1x.
Step 14	**security web-auth** **Example:** `Device(config-wlan)# security web-auth`	Enables web authentication for WLAN.
Step 15	**security web-auth authentication-list** *authenticate-list-name* **Example:** `Device(config-wlan)# security web-auth authentication-list WEBAUTH`	Enables authentication list for dot1x security.
Step 16	**security web-auth parameter-map** *parameter-map-name* **Example:** `Device(config-wlan)# security web-auth parameter-map ISE-Ext-Webauth_IP`	Configures the parameter map. **Note** If parameter map is not associated with a WLAN, the configuration is considered from the global parameter map.

	Command or Action	Purpose
Step 17	**end** **Example:** `Device(config-wlan)# end`	Returns to privileged EXEC mode.

Configuring EWA with Multiple Web Servers and/or Ports Different than Default (80/443)

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**ip access-list extended** *name* **Example:** `Device(config)# ip access-list extended preauth_ISE_Ext_WA`	Defines an extended IPv4 access list using a name, and enters access-list configuration mode.
Step 3	*access-list-number* **permit tcp any host** *external_web_server_ip_address1* **eq** *port-number* **Example:** `Device(config)# 10 permit tcp any host 192.168.0.98 eq 8443`	Permits access from any host to the external web server port number 8443.
Step 4	*access-list-number* **permit tcp any host** *external_web_server_ip_address2* **eq** *port-number* **Example:** `Device(config)# 10 permit tcp any host 192.168.0.99 eq 8443`	Permits access from any host to the external web server port number 8443.
Step 5	*access-list-number* **permit udp any any eq domain** **Example:** `Device(config)# 20 permit udp any any eq domain`	Permits DNS UDP traffic.
Step 6	*access-list-number* **permit udp any any eq bootpc** **Example:** `Device(config)# 30 permit udp any any eq bootpc`	Permits DHCP traffic.

	Command or Action	Purpose
Step 7	*access-list-number* **permit udp any any eq bootps** **Example:** `Device(config)# 40 permit udp any any eq bootps`	Permits DHCP traffic.
Step 8	*access-list-number* **permit tcp host** *external_web_server_ip_address1* **eq** *port_number* **any** **Example:** `Device(config)# 50 permit tcp host 192.168.0.98 eq 8443 any`	Permits the access from the external web server port 8443 to any host.
Step 9	*access-list-number* **permit tcp host** *external_web_server_ip_address2* **eq** *port_number* **any** **Example:** `Device(config)# 50 permit tcp host 192.168.0.99 eq 8443 any`	Permits the access from the external web server port 8443 to any host.
Step 10	*access-list-number* **permit tcp any any eq domain** **Example:** `Device(config)# 60 permit tcp any any eq domain`	Permits the DNS TCP traffic.
Step 11	*access-list-number* **deny ip any any** **Example:** `Device(config)# 70 deny ip any any`	Denies all the other traffic.
Step 12	**wlan** *wlan-name wlan-id ssid* **Example:** `Device(config)# wlan EWLC3-GUEST 3 EWLC3-GUEST`	Creates the WLAN.
Step 13	**ip access-group web** *name* **Example:** `Device(config-wlan)# ip access-group web preauth_ISE_Ext_WA`	Configures the IPv4 WLAN web ACL. The variable *name* specifies the user-defined IPv4 ACL name.
Step 14	**end** **Example:** `Device(config-wlan)# end`	Returns to privileged EXEC mode.

Configuring Wired Guest EWA with Multiple Web Servers and/or Ports Different than Default (80/443)

Before you begin

You cannot assign a manual ACL to a wired guest LAN configuration. The workaround is to use the bypass ACL in the global parameter map.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**ip access-list extended** *name* **Example:** `Device(config)# ip access-list extended BYPASS_ACL`	Defines an extended IPv4 access list using a name, and enters access-list configuration mode.
Step 3	*access-list-number* **deny ip any host** *hostname* **Example:** `Device(config)# 10 deny ip any host 192.168.0.45`	Allows the traffic to switch centrally.
Step 4	*access-list-number* **deny ip any host** *hostname* **Example:** `Device(config)# 20 deny ip any host 4.0.0.1`	Allows the traffic to switch centrally.
Step 5	**parameter-map type webauth global** **Example:** `Device(config)# parameter-map type webauth global`	Creates a parameter map and enters parameter-map webauth configuration mode.
Step 6	**webauth-bypass-intercept** *name* **Example:** `Device(config-params-parameter-map)# webauth-bypass-intercept BYPASS_ACL`	Creates a WebAuth bypass intercept using the ACL name. **Note** You cannot apply a manual ACL to the wired guest profile and configure an external web authentication with multiple IP addresses or different ports. The workaround is to use the bypass ACL for wired guest profile.
Step 7	**end** **Example:** `Device(config-params-parameter-map)# end`	Returns to privileged EXEC mode.

Authentication for Sleeping Clients

Information About Authenticating Sleeping Clients

Clients with guest access that have had successful web authentication are allowed to sleep and wake up without having to go through another authentication process through the login page. You can configure the duration for which sleeping clients should be remembered for before reauthentication becomes necessary. The valid range is 10 minutes to 43200 minutes, with the default being 720 minutes. You can also configure this duration on WebAuth parameter map that is mapped to a WLAN. Note that the sleeping client timer comes into effect due to instances such as idle timeout, session timeout, disabling of the WLAN, and the AP being nonoperational.

This feature is supported in the following FlexConnect scenario: local switching and central authentication.

Caution If the MAC address of a client that goes to sleep mode is spoofed, the fake device such as a laptop can be authenticated.

Mobility Scenarios

Following are some guidelines in a mobility scenario:

- L2 roaming in the same subnet is supported.
- Anchor sleeping timer is applicable.
- The sleeping client information is shared between multiple autoanchors when a sleeping client moves from one anchor to another.

A sleeping client does not require reauthentication in the following scenarios:

- Suppose there are two controller s in a mobility group. A client that is associated with one controller goes to sleep and then wakes up and gets associated with the other controller .
- Suppose there are three controller s in a mobility group. A client that is associated with the second controller that is anchored to the first controller goes to sleep, wakes up, and gets associated with the third controller .
- A client sleeps, wakes up and gets associated with the same or different export foreign controller that is anchored to the export anchor.

Restrictions on Authenticating Sleeping Clients

- The sleep client feature works only for WLAN configured with WebAuth security.
- You can configure the sleeping clients only on a per WebAuth parameter-map basis.
- The authentication of sleeping clients feature is supported only on WLANs that have Layer 3 security enabled.
- With Layer 3 security, the Authentication, Passthrough, and On MAC Filter failure web policies are supported. The Conditional Web Redirect and Splash Page Web Redirect web policies are not supported.

- The central web authentication of sleeping clients is not supported.
- The authentication of sleeping clients feature is not supported on guest LANs and remote LANs.
- A guest access sleeping client that has a local user policy is not supported. In this case, the WLAN-specific timer is applied.

Configuring Authentication for Sleeping Clients (GUI)

Procedure

Step 1 Choose **Configuration** > **Security** > **Web Auth**.

Step 2 In the **Webauth Parameter Map** tab, click the parameter map name. The **Edit WebAuth Parameter** window is displayed.

Step 3 Select **Sleeping Client Status** check box.

Step 4 Click **Update & Apply to Device**.

Configuring Authentication for Sleeping Clients (CLI)

Procedure

	Command or Action	Purpose
Step 1	**[no] parameter-map type webauth** {*parameter-map-name* | **global**} **Example:** `Device(config)# parameter-map type webauth global`	Creates a parameter map and enters parameter-map webauth configuration mode.
Step 2	**sleeping-client** [**timeout** *time*] **Example:** `Device(config-params-parameter-map)# sleeping-client timeout 100`	Configures the sleeping client timeout to 100 minutes. Valid range is between 10 minutes and 43200 minutes. **Note** If you do not use the timeout keyword, the sleeping client is configured with the default timeout value of 720 minutes.
Step 3	**end**	Exits parameter-map webauth configuration mode and returns to privileged EXEC mode.
Step 4	(Optional) **show wireless client sleeping-client** **Example:** `Device# show wireless client sleeping-client`	Shows the MAC address of the clients and the time remaining in their respective sessions.

	Command or Action	Purpose
Step 5	(Optional) **clear wireless client sleeping-client** [**mac-address** *mac-addr*] **Example:** `Device# clear wireless client sleeping-client mac-address 00e1.e1e1.0001`	• **clear wireless client sleeping-client**—Deletes all sleeping client entries from the sleeping client cache. • **clear wireless client sleeping-client mac-address** *mac-addr*—Deletes the specific MAC entry from the sleeping client cache.

Sleeping Clients with Multiple Authentications

Mobility Support for Sleeping Clients

From Release 17.1.1 onwards, mobility support for guest and nonguest sleeping clients.

Supported Combinations of Multiple Authentications

Multiple authentication feature supports sleeping clients configured in the WLAN profile.

The following table outlines the supported combination of multiple authentications:

Table 3: Supported Combinations of Multiple Authentications

Layer 2	Layer 3	Supported
MAB	LWA	Yes
MAB Failure	LWA	Yes
Dot1x	LWA	Yes
PSK	LWA	Yes

Configuring Sleeping Clients with Multiple Authentications

Configuring WLAN for Dot1x and Local Web Authentication

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 2	**wlan** *profile-name wlan-id SSID_name* **Example:** `Device(config)# wlan wlan-test 3 ssid-test`	Enters WLAN configuration submode. • *profile-name* - Profile name of the configured WLAN. • *wlan-id* - Wireless LAN identifier. Range is from 1 to 512. • *SSID_Name* - SSID, which can contain up to 32 alphanumeric characters.
Step 3	**security dot1x authentication-list** *auth-list-name* **Example:** `Device(config-wlan)# security dot1x authentication-list default`	Enables security authentication list for dot1x security. The configuration is similar for all dot1x security WLANs.
Step 4	**security web-auth** **Example:** `Device(config-wlan)# security web-auth`	Configures web authentication.
Step 5	**security web-auth authentication-list** *authenticate-list-name* **Example:** `Device(config-wlan)# security web-auth authentication-list default`	Enables authentication list for dot1x security.
Step 6	**security web-auth parameter-map** *parameter-map-name* **Example:** `Device(config-wlan)# security web-auth parameter-map global`	Maps the parameter map. **Note**: If the parameter map is not associated with a WLAN, the configuration is considered from the global parameter map.
Step 7	**no shutdown** **Example:** `Device(config-wlan)# no shutdown`	Enables WLAN.

Configuring a WLAN for MAC Authentication Bypass and Local Web Authentication

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**wlan** *profile-name wlan-id SSID_name* **Example:** `Device(config)# wlan wlan-test 3 ssid-test`	Enters WLAN configuration submode. • *profile-name* - Profile name of the configured WLAN. • *wlan-id* - Wireless LAN identifier. Range is from 1 to 512. • *SSID_Name* - SSID, which can contain up to 32 alphanumeric characters.
Step 3	**mac-filtering** *list-name* **Example:** `Device(config-wlan)# mac-filtering cat-radius`	Sets the MAC filtering parameters.
Step 4	**no security wpa akm dot1x** **Example:** `Device(config-wlan)# no security wpa akm dot1x`	Disables security AKM for dot1x.
Step 5	**no security wpa wpa2 ciphers aes** **Example:** `Device(config-wlan)# no security wpa wpa2 ciphers aes`	Disables the WPA2 cipher. **aes**—Excryption type that specifies WPA/AES support.
Step 6	**security web-auth parameter-map** *parameter-map-name* **Example:** `Device(config-wlan)# security web-auth parameter-map global`	Maps the parameter map. **Note**: If parameter map is not associated with a WLAN, the configuration is considered from the global parameter map.
Step 7	**no shutdown** **Example:** `Device(config-wlan)# no shutdown`	Enables WLAN.

Configuring a WLAN for Local Web Authentication and MAC Filtering

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**wlan** *profile-name wlan-id SSID_name* **Example:** `Device(config)# wlan wlan-test 3 ssid-test`	Enters WLAN configuration submode. • *profile-name* - Profile name of the configured WLAN. • *wlan-id* - Wireless LAN identifier. Range is from 1 to 512. • *SSID_Name* - SSID, which can contain up to 32 alphanumeric characters.
Step 3	**mac-filtering** *list-name* **Example:** `Device(config-wlan)# mac-filtering cat-radius`	Sets the MAC filtering parameters.
Step 4	**no security wpa akm dot1x** **Example:** `Device(config-wlan)# no security wpa akm dot1x`	Disables security Authenticated Key Management (AKM) for dot1x.
Step 5	**no security wpa wpa2 ciphers aes** **Example:** `Device(config-wlan)# no security wpa wpa2 ciphers aes`	Disables the WPA2 cipher. **aes**: Excryption type that specifies WPA/AES support.
Step 6	**security web-auth on-macfilter-failure** **Example:** `Device(config-wlan)# security web-auth on-macfilter-failure wlan-id`	Configures the fallback policy with MAC filtering and web authentication.
Step 7	**security web-auth parameter-map** *parameter-map-name* **Example:** `Device(config-wlan)# security web-auth parameter-map global`	Maps the parameter map. **Note**: If the parameter map is not associated with a WLAN, the configuration is considered from the global parameter map.

	Command or Action	Purpose
Step 8	**no shutdown** **Example:** `Device(config-wlan)# no shutdown`	Enables WLAN.

Configuring a PSK + LWA in a WLAN

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**wlan** *profile-name wlan-id SSID_name* **Example:** `Device(config)# wlan wlan-test 3 ssid-test`	Enters WLAN configuration submode. • *profile-name* - Profile name of the configured WLAN. • *wlan-id* - Wireless LAN identifier. Range is from 1 to 512. • *SSID_Name* - SSID, which can contain up to 32 alphanumeric characters.
Step 3	**no security wpa akm dot1x** **Example:** `Device(config-wlan)# no security wpa akm dot1x`	Disables security AKM for dot1x.
Step 4	**security web-auth** **Example:** `Device(config-wlan)# security web-auth`	Enables web authentication for a WLAN.
Step 5	**no security wpa wpa2 ciphers aes** **Example:** `Device(config-wlan)# no security wpa wpa2 ciphers aes`	Disables the WPA2 cipher. **aes**: Excryption type that specifies WPA/AES support.
Step 6	**security wpa psk set-key ascii** *ascii/hex key* **Example:** `Device(config-wlan)# security wpa psk set-key ascii 0 1234567`	Configures the preshared key on a WLAN.
Step 7	**security wpa akm psk** **Example:**	Configures PSK support.

	Command or Action	Purpose
	`Device(config-wlan)# security wpa akm psk`	
Step 8	**security web-auth authentication-list** *authenticate-list-name* **Example:** `Device(config-wlan)# security web-auth authentication-list default`	Enables the authentication list for dot1x security.
Step 9	**security web-auth parameter-map** *parameter-map-name* **Example:** `Device(config-wlan)# security web-auth parameter-map global`	Maps the parameter map. **Note**: If the parameter map is not associated with a WLAN, the configuration is considered from the global parameter map.

Configuring a Sleeping Client

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**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-name* 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 3	**sleeping client** [**timeout** *time*] **Example:** `Device(config-params-parameter-map)# sleeping-client timeout 60`	Configures the sleeping client timeout, in minutes. The available range for the *time* argument is from 10 to 43200. **Note**: If you do not use the **timeout** keyword, the sleeping client is configured with the default timeout value of 720 minutes.

Verifying a Sleeping Client Configuration

To verify a sleeping client configuration, use the following command:

```
Device# show wireless client sleeping-client
Total number of sleeping-client entries: 1

MAC Address                    Remaining time (mm:ss)
------------------------------------------------------------
2477.031b.aa18                 59:56
```