Web-Based Authentication

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

- Local web authentication , on page 1
- External Web Authentication, on page 28
- Authentication of Sleeping Client, on page 33

Local web authentication

Local web authentication is a network security mechanism that

- authenticates users through a web browser login page
- enables access control on host systems that do not run IEEE 802.1X supplicants, and
- communicates with authentication, authorization, and accounting (AAA) servers to enforce security policies.

Feature history

Feature Name	Release	Description
Built-in Captive Portal Improvement	Cisco IOS XE 17.1.1	This release introduces support for special characters in the login portal banner title and banner text. The number of characters supported on the banner text has doubled to 400. The **exec-character-bits** command has been introduced.

Presentation options for local web authentication web pages

Local web authentication intercepts HTTP sessions on Layer 2 interfaces, and in some cases, Layer 3 interfaces (with restrictions for some switch models). When users try to access the network, local web authentication displays a login page and verifies user credentials with AAA servers, granting or denying access accordingly.

Local web authentication is categorized by the location where its web pages are hosted.

- **Internal**: Uses HTML pages (login, success, fail, and expire) stored on the controller .

- **Customized**: Uses customized HTML pages (login, success, fail, and expire) downloaded onto the controller for a customized user experience.
- **External**: Uses HTML pages hosted on an external web server.

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

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

Web authentication modes

The types of web authentication differ according to the available web authentication pages.

- **Webauth**—The controller displays a page with the user name and password. Users enter valid credentials to gain network access.
- **Consent or web-passthrough**—The controller presents a policy page with the **Accept** and **Deny** buttons. Users simply click **Accept** to access the network – no credentials are required.
- **Webconsent**—This mode combines the features of Webauth and Consent. The controller displays a policy page with Accept or Deny buttons along with user name or password. Users must enter the correct credentials and click **Accept** to access the network.

Additional reference information

- You can view the webauth parameter-map information using the **show running-config** command.
- If you see tracebacks during client authentication, performance and behavior are not affected. Tracebacks may occur if Flexible Forwarding Mode (FFM) replies to Endpoint Profiler Module (EPM) for ACL application after the session is dequeued, usually because a timer expired or the session became **unauthorized**.
- Apply web authentication methods (such as consent, web consent, and webauth) using either a global or named parameter-map under WLAN (for method-type, custom, and redirect). If you do not configure a parameter-map under WLAN, the global parameter-map applies by default.
- You can configure web-based authentication on Layer 2 and layer 3 interfaces.
- When a client reaches the maximum limit of 200 HTTP connections, the system resets TCP connections and excludes your client from the network.

How local web authentication works

Summary

authentication enables secure network access for clients by prompting users to authenticate through a web login page. The process coordinates actions between the user, authentication server, and the network switch. The key components are:

- **User**: Initiates the HTTP session and enters authentication credentials.
- **Network switch**: Intercepts traffic, presents login pages, applies policies, and communicates with the authentication server.

- **Authentication server**: Verifies credentials and provides policy enforcement details.

Workflow

These are the stages of the process.

1. **Session initiation**: The user starts an HTTP session by attempting to access the network.
2. **Traffic interception and login page presentation**: The network switch intercepts the HTTP request and triggers the authorization process. It presents a login page for the user to enter their username and password.
3. **Credential submission and authentication**: After the user submits credentials, the switch forwards them to the authentication server
4. **Authentication outcome:**
   - If authentication succeeds, the switch downloads and activates the access policy assigned to the user from the server, then displays a login success page.
   - If authentication fails, the switch displays a login failure page. After a maximum number of failures, the login expired window appears, and the host is put on a watch list. Once the timeout expires, the user may try again
5. **Server non-response handling**: If the authentication server does not respond, and an AAA fail policy is in place, the switch applies the failure policy to the host and displays a login success page.
6. **Reauthentication triggers**: The switch reauthenticates a client if the host does not respond to an ARP probe (Layer 2) or does not send traffic within an idle timeout (Layer 3)

   suppress-feature-id="uabu_2960l_sw"

   .
7. **Session timeout enforcement**: The switch applies either the session timeout configured locally or provided by the server. The default local web authentication session timeout on controller is 1800 seconds from Cisco IOS XE 16.1.1 and later. The default session timeout value was infinite seconds, prior to Cisco IOS XE Denali 16.1.1.
8. **Session termination:**
   - If the terminate action is set to RADIUS, the switch sends a nonresponsive host request to the server; the server's response dictates the next action.
   - If the terminate action is set to default, the switch dismantles the session and removes the access policy.

Result

If authentication succeeds and the correct policies are applied, the user gains network access. If authentication fails, failure policies or timeout mechanisms may deny access.

Restrictions

- You cannot configure bypass authentication with the wireless web authentication feature.
- To update the redirect login URL, enable and then disable the WLAN. The redirect login URL specified in the web authentication parameter map changes when an AP rejoins.

- If authentication fails, users receive a failure page and can try to log in again. If the number of allowed attempts is exceeded, users may be excluded and receive a specific reason for the exclusion.
- Use the local web authentication feature to authenticate end users on host systems that do not run the IEEE 802.1x supplicant.

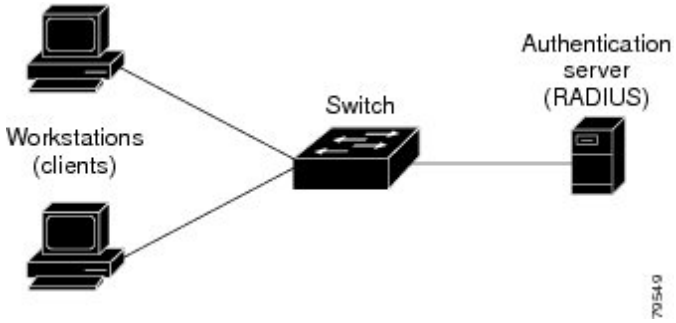
Roles of devices in local web authentication

In a local web authentication scenario, network devices assume specific roles to manage authentication and access to the LAN.

- **Client**: A device, such as a workstation, that requests network access and responds to authentication requests from the switch. The client must have an HTML browser with JavaScript enabled.
- **Authentication server**: A server that validates the identity of the client. The authentication server notifies the switch if the client is allowed or denied access to the LAN and related services.
- **Switch**: A network device that manages physical access to the network based on the authentication status of the client. The switch relays identity information and authorization responses between the client and the authentication server.

These device roles work together to ensure secure access control through local web authentication processes.

Figure 1: Local web authentication device roles



Banner messages for local web authentication

Local web authentication banners provide visual feedback to users during authentication on switches. The banners can display default or customized messages. They may also include branding or additional information on the login and result screens.

Default banner messages

When web authentication is enabled, one of these default messages appears on both the login screen and the authentication result pop-up page:

- Authentication Successful
- Authentication Failed
- Authentication Expired

Commands to configure local web authentication banner

You can configure the local web authentication banner using the new style (Session-aware) CLI mode.

Use the command in the global configuration mode:

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

You can add a custom message, such as a switch, router, or company name, to the banner using the command:

```
Device(config)# parameter map type webauth global
Device(config-params-parameter-map)# banner text <text>
```

You can add a logo or text file to the banner using the command:

```
Device(config)# parameter map type webauth global
Device(config-params-parameter-map)# banner text <filepath>
```

Banner examples

Figure 2: Customized web banner: Authentication Successful

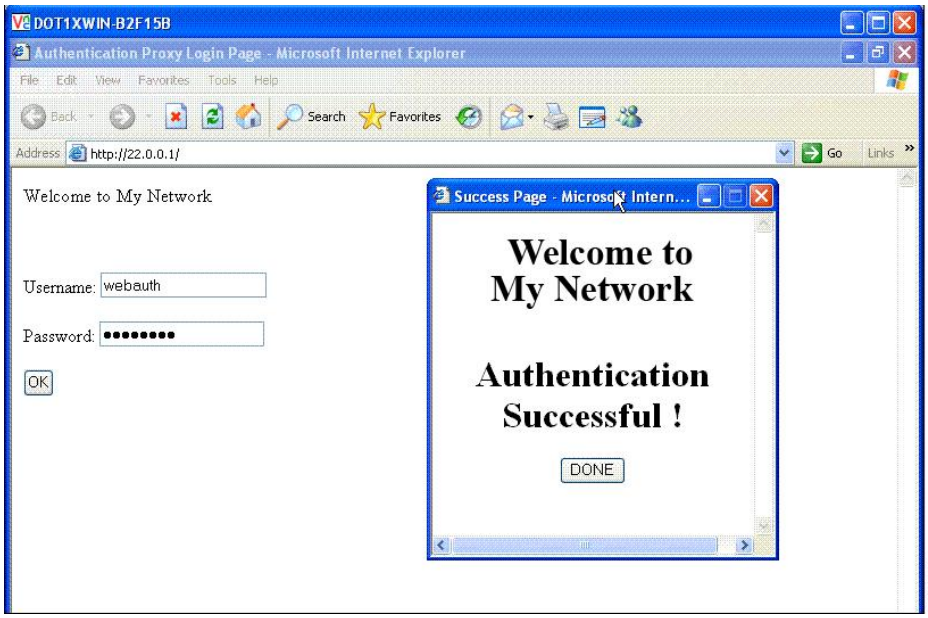
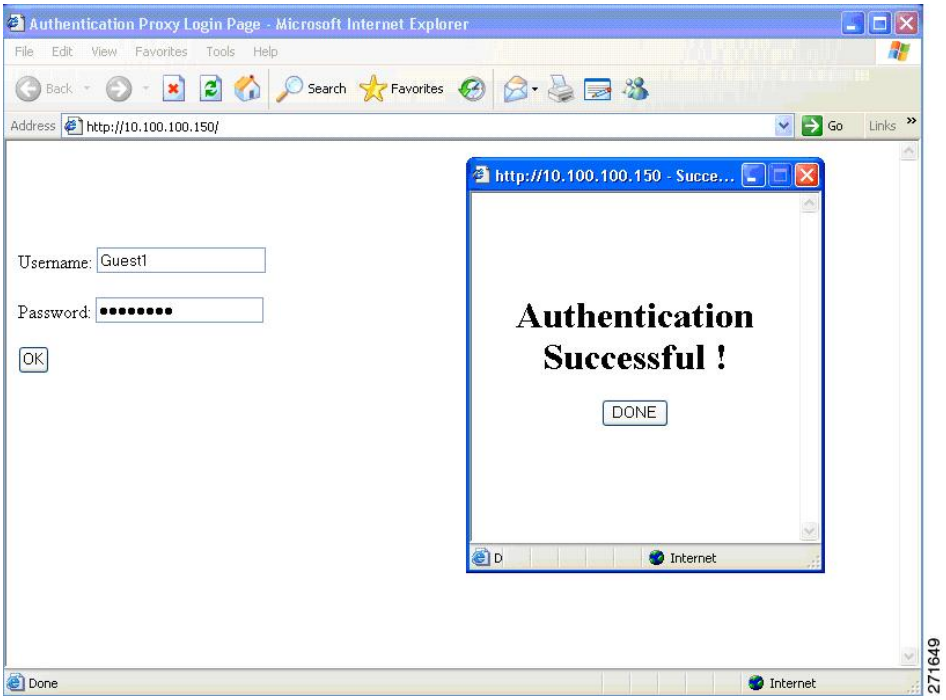


Figure 3: Login screen with no banner



Banner usage and behavior

- The default banners are *Cisco Systems* and *Switch host-name Authentication*. These banners appear on the login page. The *Cisco Systems* banner also appears on the authentication result pop-up page.
- The default banners appear unless custom banners are configured.
- If you do not enable a banner, only username and password dialog boxes appear on the web authentication login screen. You will not see a banner when you log into the switch.

Authentication states in customized local web authentication

During local web authentication, the switch's internal HTTP server hosts four HTML pages that communicate authentication states to the client.

You can replace the default internal HTML pages or specify a URL that redirects users after authentication. This URL replaces the internal Success page.

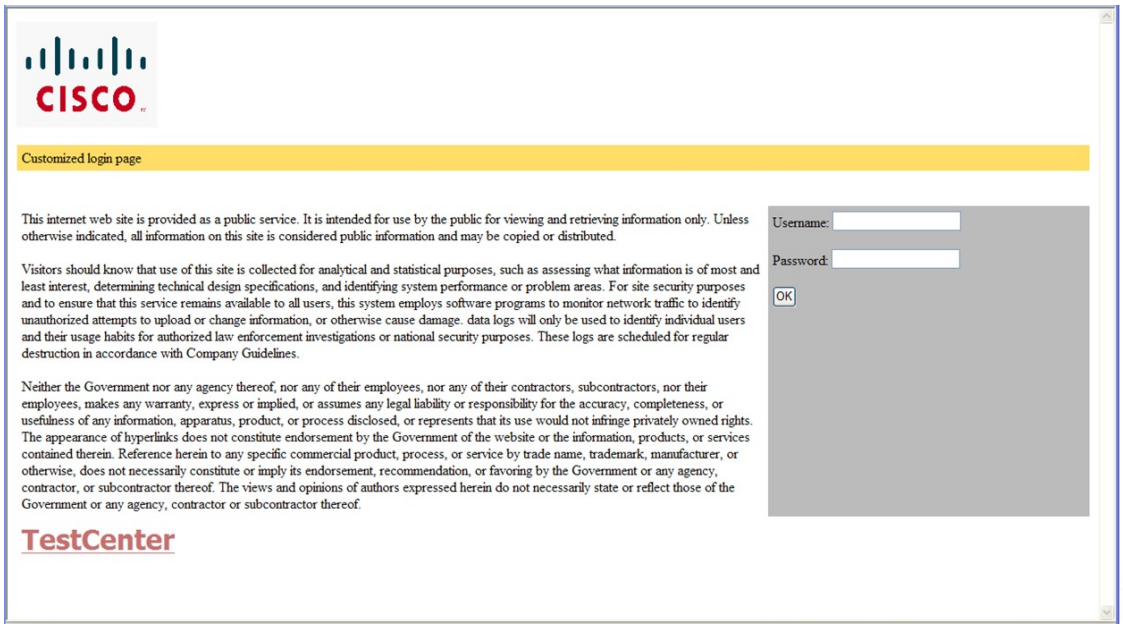
The four authentication pages and states are:

- Login: Credentials are requested from the client.
- Success: The client has authenticated successfully.
- Fail: Authentication attempt failed.
- Expire: The session expires after excessive failures.

You must configure all four pages. You can specify text or add a logo to each of the four pages.

Banner examples

Figure 4: Customizable Authentication Page



Best practices for customizing web authentication pages

- Add appropriate text to the banner and login pages as needed.
- Always include a valid HTML redirect command in the success page to redirect users to a specific URL after login
- Ensure the URL string is well-formed (for example, "http://www.cisco.com") to avoid browser errors.
- If you configure web pages for HTTP authentication, include the appropriate HTML commands. For example, HTML commands to set the page time out, to set a hidden password, or to confirm that the same page is not submitted twice.
- You can copy the configured web pages to the switch boot flash or the flash.
- The login page can reside on one flash device, while the success and failure pages can be stored on another flash device
- You must configure all four pages.
- All logo files—including image, flash, audio, video, and similar file types—stored in the system directory must use *web_auth_<filename>* as the file name. System directory examples are flash, disk0, or disk.
- You can copy the configured web pages to the switch boot flash or the flash.

Restrictions for customizing web authentication pages

- The banner page has no effect if it is configured together with a web authentication page.
- When the configured login form is enabled, the CLI command for redirecting users to a specific URL is unavailable. Configure redirection in the web page.

- If you enter the CLI command to redirect users to a specific URL after authentication and then configure web pages, the redirect command does not take effect.
- The configured authentication proxy feature supports both HTTP and SSL.

Guidelines for configuring a redirection URL for Successful Login page

- If you enable the custom authentication proxy web pages feature, you cannot use the redirection URL feature in the CLI. To redirect users after login, configure redirection in the custom login success page.
- If you enable the redirection URL feature, the configured authorization proxy banner is not be used.
- To remove the specification of a redirection URL, use the **no** form of the command.
- If a redirection URL is required after successful authentication, it must begin with a valid protocol prefix (such as http://) followed by the URL. If http:// is omitted, the browser might show a page not found error or similar issue

How to configure local web authentication

Configure default 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	Enabled

Configure AAA authentication (GUI)

Set up AAA authentication to control network access using the GUI.

Use this task to define and assign authentication methods and server groups for device access.

Before you begin

Confirm that server groups are configured if you plan to use them.

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 In the **Group** Type drop-down list, select either a group of servers as your access server or a local server to authenticate access.

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**.

The controller is now configured with AAA authentication, and users are authenticated using the chosen methods.

Configure AAA authentication (CLI)

Enable authentication, authorization, and accounting (AAA) on the device to manage login methods and network access.

AAA centralizes user authentication for device access and network services. Select a named method list or the default list, depending on your VTY line configuration.

If no method list is configured under VTY lines, add the default method list to the AAA configuration."

```
line vty 0 4
 authorization commands 15 abc
aaa authorization commands 15 abc group tacacs+
```

If a method-list is not configured under VTY lines, you must add the default method list to the AAA configuration:

```
line vty 0 4
 aaa authorization commands 15 default group tacacs+
```

Note Use the default list for AAA authorization if you plan to use features such as dACL.

Before you begin

Ensure that the TACACS+ server address and group name are available.

Procedure

Step 1 Enable AAA functionality.

Example:

```
Device
(config)# aaa new-model
```

Step 2 Define the list of authentication methods at login using the **aaa authentication login {default |** *named_authentication_list*} **group** *AAA_group_name* command.

Example:

```
Device(config)# aaa authentication login default group AAA_group_name
```

named_authentication_list refers to any name that is not greater than 31 characters.

AAA_group_name refers to the server group name. Define the **server_name** at the beginning of the configuration.

Step 3 Create an authorization method list for web-based authorization using the **aaa authorization network {default |** *named_authorization_list*} **group** *AAA_group_name* command.

Example:

```
Device(config)# aaa authorization network default group group1
```

Step 4 Specify an AAA server.

Example:

```
Device(config)# tacacs server yourserver
```

Step 5 Configure the IP address for the TACACS server.

Example:

```
Device(config-server-tacacs)# address ipv4 10.0.1.12
```

Step 6 Specify a AAA server.

Example:

```
Device(config)# tacacs-server host 10.1.1.1
```

Configure the HTTP or HTTPS server (GUI)

Set up HTTP and HTTPS access to enable secure web-based management of the device.

HTTP provides basic web access; HTTPS secures connections with SSL encryption. You can configure access ports, authentication, trust points, and session policies to meet your security requirements.

Before you begin

Confirm necessary certificates are available if using HTTPS trust points.

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. Data exchanged over a secure HTTP connection is encrypted before transmission across the Internet. SSL encryption over HTTP enables secure configuration of the switch through 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. Enter a value between one to 600 seconds.

Step 8 Enter the number of minutes of inactivity allowed before the session times out. Enter a value between 180 and 1200 seconds.

Step 9 Enter the server life time in seconds. Enter a value between one and 86400 seconds.

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

Step 11 Save the configuration.

The device is configured for HTTP or HTTPS access according to your specified settings.

Configure the HTTP server (CLI)

Enable HTTP or HTTPS server functionality on your device to support local web authentication.

Local web authentication requires the HTTP server to be enabled on the controller.

You can configure both HTTP and HTTPS servers to support device management and user authentication.

Both HTTP and HTTPS servers can be configured to support device management and user authentication.

Note that some browsers, such as the Apple psuedo-browser, do not open if you configure only the **ip http secure-server** command. You should also configure the **ip http server** command.

Procedure

Step 1 Enable the HTTP server. The local web authentication feature uses the HTTP server to communicate with the hosts for user authentication.

Example:

```
Device(config)# ip http server
```

Step 2 Enable HTTPS.

Example:

```
Device(config)# ip http secure-server
```

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 3 Return to privileged EXEC mode.

Example:

```
Device(config)# end
```

The controller is now configured with the HTTP or HTTPS or both. This enables local web authentication and secure browser access for users.

Create a parameter map (GUI)

Define criteria-based policies to control device and user access within the local policy framework.

Use this task to create a parameter map by specifying match criteria and associating service templates. This enables dynamic policy application based on device type, user role, and other attributes.

Before you begin

Ensure you have the required policy details and service templates ready.

Procedure

Step 1 Choose **Configuration** > **Security** > **Local Policy**.
Step 2 Click **Add**.
Step 3 Click **Policy Map**.
Step 4 Enter **Policy Map Name**.
Step 5 In the **Match Criteria List** settings, click **Add**.
Step 6 In the **Add Match Criteria** settings, choose the service template from the **Service Template** drop-down list.
Step 7 Choose the filters from **Device Type**, **User Role**, **User Name**, **OUI**, and **MAC Address** drop-down lists.
Step 8 Click **Add Criteria**.
Step 9 Click **Apply to Device**.

The parameter map is created and applied, enabling policy enforcement based on the selected criteria.

Creating parameter maps

Configure local web authentication (GUI)

Enable local web authentication for your network using GUI.

Use this task to set authentication parameters, banner messages, and related settings for web authentication.

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 timeout duration. The timer expires if the user does not enter valid credentials on 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 the status of **WebAuth Intercept HTTPS**, **Captive Bypass Portal**, and **Watch List Enable** as required.

Step 12 In the **Watch List Expiry Timeout** field, enter the time in seconds after which the watch list should time out.

Step 13 Set the status for **Disable Success Window**, **Disable Logout Window**, and **Login Auth Bypass for FQDN**, as required.

Step 14 Check the **Sleeping Client Status** check box to enable authentication of sleeping clients. Specify the **Sleeping Client Timeout** in minutes. The valid range is 10 to 43,200 minutes.

Step 15 Click the **Advanced** tab.

Step 16 In the **Redirect for log-in** field, enter the name of the external server to which to send the login request.

Step 17 In the **Redirect On-Success** field, enter the name of the external server to redirect after a successful login.

Step 18 In the **Redirect On-Failure** field, enter the name of the external server to redirect after a login failure.

Step 19 To configure external local web authentication, perform these steps:

a) Under **Redirect to External Server** in the **Redirect Append for AP MAC Address** field, enter the AP MAC address.
b) In the **Redirect Append for Client MAC Address** field, enter the client MAC address.
c) In the **Redirect Append for WLAN SSID** field, enter the WLAN SSID.
d) In the **Portal IPV4 Address** field, enter the IPv4 address of the portal to send redirects.
e) In the **Portal IPV6 Address** field, enter the IPv6 address of the portal to send redirects if you are using an IPv6 address.

Step 20 To configure customized local web authentication, perform these steps:

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

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

Step 21 Click **Update & Apply**.

Local web authentication is enabled with your specified settings.

Configure internal local web authentication (CLI)

Set up internal local web authentication on your device using CLI.

Procedure

In the global configuration mode, create a parameter map

Example:

```
Device(config)# parameter-map type webauth sample
```

Creates the parameter map.

The name of the parameter map must not exceed 99 characters.

You have configured internal local web authentication.

The device prompts clients to authenticate through the web portal. After authentication, the device grants full network access.

Configure the customized local web authentication (CLI)

Configure custom local web authentication workflow on a Cisco device by specifying custom portal pages using CLI.

Perform this procedure to offer users a personalized login, success, and failure experience during web authentication.

Before you begin

- Ensure you have valid HTML files for login, success, expired, and failure pages uploaded to device storage.
- Configure a virtual IP in the global parameter map to use the customized web authentication bundle.

Procedure

Step 1 Configure the webauth type parameter.

Example:

```
Device(config)# parameter-map type webauth parameter-map-name
```

Configure a virtual IP in the global parameter map so the device uses the customized web authentication bundle.

Step 2 Configures webauth sub-types using the **type** {**authbypass** | **consent** | **webauth** | **webconsent**} command.

Example:

```
Device(config-params-parameter-map)# type webauth
```

The types are **passthru**, **consent**, **webauth**, or **webconsent**.

Step 3 Configure the customized login page using the **custom-page login device** *html-filename* command.

Example:

```
Device(config-params-parameter-map)# custom-page login device bootflash:login.html
```

Step 4 Configure the customized login expiry page using the **custom-page login expired device** *html-filename* command.

Example:

```
Device(config-params-parameter-map)# custom-page login expired device
bootflash:loginexpired.html
```

Step 5 Configure the customized login success page using the **custom-page success device** *html-filename* command.

Example:

```
Device(config-params-parameter-map)# custom-page success device bootflash:loginsuccess.html
```

Step 6 Configure the customized login failure page using the **custom-page failure device** *html-filename* command.

Example:

```
Device(config-params-parameter-map)# custom-page failure device bootflash:loginfail.html
```

You have configured the customized local web authentication portal. Users can see the specified custom pages during authentication.

Configure the external local web authentication (CLI)

Enable external local web authentication on your device by configuring the required CLI parameters.

Use this procedure to set up external local web authentication settings, including parameter maps and redirect URLs, using the CLI.

Before you begin

Gather the parameter map name and redirect URLs.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure the webauth type parameter.

Example:

```
Device(config)# parameter-map type webauth sample
```

Step 3 Configure web authentication subtypes by using the **type** {**authbypass** | **consent** | **webauth** | **webconsent**} command.

Example:

```
Device(config-params-parameter-map)# type webauth
```

Step 4 Configure the redirect URL for the login, failure, and success pages using the **redirect** [**for-login** | **on-failure** | **on-success**] *URL* command.

Example:

```
Device(config-params-parameter-map)# redirect for-login
http://www.cisco.com/login.html
```

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 Configure the external portal IPv4 address using the **redirect portal** {**ipv4** | **ipv6**} *ip-address* command.

Example:

```
Device(config-params-parameter-map)# redirect portal ipv4 23.0.0.1
```

When using FQDN, use an IP address associated with the domain. If the domain resolves to more than one IP address, use the FQDN URL instead of an individual IP address.

Step 6 Return to privileged EXEC mode.

Example:

```
Device(config-params-parameter-map)# end
```

You have configured external local web authentication. Your device redirects users to the correct portal for authentication events, based on your parameter map and settings.

Configure the web authentication WLANs

Configure a WLAN with web authentication security using CLI. Map authentication lists and parameter maps using the CLI.

Perform this task to set up a WLAN that uses web authentication, specifying key identifiers and settings through the CLI on your network device.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Enter the WLAN configuration sub-mode.

Example:

```
Device(config)# wlan wlan-name wlan-id SSID-name
```

Step 3 Disable WPA security.

Example:

```
Device(config-wlan)# no security wpa
```

Step 4 Enable web authentication for WLAN.

Example:

```
Device(config-wlan)# security web-auth
```

Step 5 Enable web authentication for WLAN using the **security web-auth** {**authentication-list** *authentication-list-name* | **parameter-map** *parameter-map-name*} command.

Example:

```
Device(config-wlan)# security web-auth authentication-list webauthlistlocal
Device(config-wlan)# security web-auth parameter-map sample
```

Here is a description of some of the parameters.

- **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 can 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 Return to privileged EXEC mode.

Example:

```
Device(config-wlan)# end
```

The WLAN is configured with web authentication security and the desired authentication and parameter settings.

Configure pre-auth web authentication ACL (GUI)

Map a pre-authentication access control list (ACL) to a WLAN to control network access for users before authentication.

Use this procedure to enforce security policies by restricting network access to specified resources while users are in the pre-authenticated state on a wireless LAN.

Before you begin

Ensure that you have configured an 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 map to the WLAN.

Step 6 Click **Update & Apply to Device**.

When you connect to this WLAN, you must adhere to the access restrictions defined by the ACL until authentication is complete.

Configure pre-authentication web authentication ACL (CLI)

Set up a pre-authentication web authentication access control list (ACL) and apply it to a WLAN using CLI commands. This configuration ensures only authorized traffic passes through before web authentication occurs.

Use this task to control the network traffic permitted before clients authenticate on a WLAN. The ACL defines which clients and protocols are permitted through during the web authentication process.

Before you begin

- Gather the required network and client information.
- Determine the ACL rules (permit or deny) and relevant addresses.

Procedure

Step 1 Enters global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Create an ACL list using the **access-list** *access-list-number* {**deny** | **permit**} *hostname source-wildcard-bits* command.

Example:

```
Device(config)# access-list 2 deny your_host 10.1.1.1 log
```

The *access-list-number* is a decimal number from one 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.

Note
The AP forwards traffic if it matches any of the ACLs applied in the pre-authentication ACLs. Use bidirectional rules for clients in web authentication pending state and for downstream traffic.

Step 3 Create the WLAN.

Example:

```
Device(config)# wlan mywlan 34 mywlan-ssid
```

profile-name is the WLAN name which can contain up to 32 alphanumeric characters.

wlan-id is the wireless LAN identifier. The valid range is from one to 512.

ssid-name is the SSID which can contain 32 alphanumeric characters.

Step 4 Map the ACL to the web auth WLAN.

Example:

```
Device(config-wlan)# ip access-group web name
```

access-list-name is the IPv4 ACL name or ID.

Step 5 Return to privileged EXEC mode.

Example:

```
Device(config-wlan)# end
```

The ACL is configured and mapped to the WLAN. Only traffic that matches the permitted criteria passes before web authentication.

Configure the maximum web authentication request retries

Set the maximum number of web authentication request retries to define how many times the system attempts authentication before it stops.

Configure this setting to adjust the tolerance for failed web authentication attempts, which can improve network security and user experience.

Before you begin

Ensure you are in global configuration mode on your device.

Procedure

Step 1 Configure the maximum web authentication request retries.

Example:

```
Device(config)# wireless security web-auth retries  number
```

number is the maximum number of web authentication request retries. The valid range is 0-20.

Step 2 Return to privileged EXEC mode.

Example:

```
Device(config)# end
```

You have configured the specified maximum number of web authentication request retries.

Configure a local banner in web authentication page (GUI)

Present a custom banner to users on the web authentication page to meet organizational messaging or compliance requirements.

Use this task to configure a banner that will display on the login page for web authentication sessions.

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**.

The specified banner displays on the web authentication page as configured.

Configure a local banner in web authentication page (CLI)

Present a custom banner to users on the Web Authentication page to meet organizational messaging or compliance requirements.

Use this task to configure a banner that displays on the login page for web authentication sessions.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure the web authentication parameters. Enters the parameter map configuration mode.

Example:

```
Device(config)# parameter-map type webauth param-map
```

Step 3 Enable the local banner using the **banner** [*file* | *banner-text* | *title*] command.

Example:

```
Device(config-params-parameter-map)# banner http C My Switch C
```

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 Return to privileged EXEC mode.

Example:

```
Device(config-params-parameter-map)# end
```

The specified banner displays on the web authentication page as configured.

Configure Type WebAuth, Consent, or both

Configure a parameter map to use webauth type, consent, or webconsent login types.

Use this configuration to enable flexible guest access authentication for users on supported devices.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure the Webauth type parameter.

Example:

```
Device(config)# parameter-map type webauth webparalocal
```

Step 3 Configure webauth type to Consent. You can configure the type as Webauth, Consent, or both (Webconsent).

Example:

```
Device(config-params-parameter-map)# type consent
```

Step 4 Return to privileged EXEC mode.

Example:

```
Device(config-params-parameter-map)# end
```

Step 5 Display the configuration details.

Example:

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

The parameter map is configured to use Webauth, Consent, or Webconsent types as specified and is ready for client authentication.

Configure preauthentication ACL

Set up a preauthentication access control list (ACL) to restrict network access prior to client authentication.

Use this task when you need to limit which resources clients can reach before they have authenticated on a WLAN.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Enter the WLAN you want to configure.

Example:

```
Device(config)# wlan ramban
```

Step 3 Disable the WLAN.

Example:

```
Device(config-wlan)# shutdown
```

Step 4 Configure the ACL to apply before authentication.

Example:

```
Device(config-wlan)# ip access-group web preauthrule
```

Step 5 Enable the WLAN again.

Example:

```
Device(config)# no shutdown
```

Step 6 Return to privileged EXEC mode.

Example:

```
Device(config-wlan)# end
```

Step 7 Display the configuration details for the WLAN

Example:

```
Device# show wlan name wlan-name
```

The preauthentication ACL is now active on the WLAN, restricting client access prior to authentication.

Configuration examples for local web authentication

Example: obtain a web authentication certificate

This example shows how to obtain web authentication certificate.

```
DeviceDevice# configure terminal
Device(config)# crypto pki import cert pkcs12 tftp://10.1.0.100/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: display 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: choose 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 wpa2
 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 CWA 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://10.1.0.100/login.html
redirect portal ipv4 10.1.0.100
security web-auth parameter-map rasagna-auth-map
security web-auth parameter-map test
```

Example: choose 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://10:1:1::100/login.html
redirect portal ipv6 10:1:1::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://10.1.0.100/login.html
 redirect portal ipv4 10.1.0.100
 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: configure 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: configure 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://10.1.0.100/login.html
 redirect portal ipv4 10.1.0.100
```

Verify web authentication type

To verify the web authentication type, run the 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 : 10.0.2.1.
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

Configure EWA with single WebAuth server address and default ports (80/443) (CLI)

Configure External Web Authentication (EWA) on your device to use a single WebAuth server address with default ports (80 or 443).

Use this procedure to redirect guest WLAN clients to a specific WebAuth portal using default HTTP or HTTPS ports.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Define the authentication method at login.

Example:

```
Device(config)# aaa authentication login WEBAUTH local
```

Step 3 Create the parameter map using the **parameter-map type webauth** *parameter-map-name* command.

Example:

```
Device(config)# parameter-map type webauth ISE-Ext-Webauth_IP
```

The *parameter-map-name* must not exceed 99 characters.

Step 4 Configure the webauth type parameter.

Example:

```
Device(config-params-parameter-map)# type webauth
```

Step 5 Configure the URL string for redirect during login using the **redirect for-login** *URL-String*command.

Example:

```
Device(config-params-parameter-map)#  redirect for-login
https://192.168.0.98:443/portal/PortalSetup.action?portal=ad64b062-1098-11e7-8591-005056891b52
```

Step 6 Configure the external portal IPv4 address.

Example:

```
Device(config-params-parameter-map)# redirect portal ipv4 192.168.0.98
```

Step 7 Return to global configuration mode.

Example:

```
Device(config-params-parameter-map)# exit
```

Step 8 Configure a WLAN using the **wlan** *wlan-name wlan-id SSID-name* command.

Example:

```
Device(config)#  wlan EWLC3-GUEST 3 EWLC3-GUEST
```

Step 9 Disable adaptive 11r.

Example:

```
Device(config-wlan)# no security ft adaptive
```

Step 10 Disable WPA security.

Example:

```
Device(config-wlan)# no security wpa
```

Step 11 Disable WPA2 security.

Example:

```
Device(config-wlan)# no security wpa wpa2
```

Step 12 Disable WPA2 ciphers for AES.

Example:

```
Device(config-wlan)# no security wpa wpa2 ciphers aes
```

Step 13 Disable security AKM for dot1x.

Example:

```
Device(config-wlan)# no security wpa akm dot1x
```

Step 14 Enable web authentication for WLAN.

Example:

```
Device(config-wlan)# security web-auth
```

Step 15 Enable authentication list for dot1x security using the **security web-auth authentication-list** *authenticate-list-name* command.

Example:

```
Device(config-wlan)# security web-auth authentication-list WEBAUTH
```

Step 16 Configure the parameter map using the **security web-auth parameter-map** *parameter-map-name* command.

Example:

```
Device(config-wlan)# security web-auth parameter-map ISE-Ext-Webauth_IP
```

Note
If parameter map is not associated with a WLAN, the configuration uses the global parameter map.

Step 17 Return to privileged EXEC mode.

Example:

```
Device(config-wlan)# end
```

Once enabled for the configured WLAN, web authentication redirects clients to the specified WebAuth portal using default ports.

Configure EWA with multiple web servers and/or ports different than default (80/443)

Configure an External Web Authentication (EWA) workflow to support multiple web servers and custom port numbers using CLI.

Enable guest access using EWA if there are multiple external web servers or if web authentication must use ports other than the default ports of 80 or 443.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Define an extended IPv4 access list using a name, and enters access-list configuration mode.

Example:

```
Device(config)# ip access-list extended preauth_ISE_Ext_WA
```

Step 3 Permit access from any host to the first external web server port number 8443.

Example:

```
Device(config)# 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
```

Step 4 Permit access from any host to the second external web server port number 8443.

Example:

```
Device(config)# 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
```

Step 5 Permit DNS UDP traffic.

Example:

```
Device(config)# access-list-number permit udp any eq domain
```

Example:

```
Device(config)# 20 permit udp any any eq domain
```

Step 6 Permit DHCP traffic using the **bootpc**.

Example:

```
Device(config)# 30 permit udp any any eq bootpc
```

Example:

```
Device(config)# access-list-number permit udp any any eq bootpc
```

Step 7 Permit DHCP traffic using **bootps**.

Example:

```
Device(config)# access-list-number permit udp any any eq bootps
```

Example:

```
Device(config)# 40 permit udp any any eq bootps
```

Step 8 Permit the access from the first external web server port 8443 to any host.

Example:

```
Device(config)# access-list-number permit tcp any host external_web_server_ip_address1 eq
port-number any
```

Example:

```
Device(config)# 50 permit tcp host 192.168.0.98 eq 8443 any
```

Step 9 Permit the access from the second external web server port 8443 to any host.

Example:

```
Device(config)# access-list-number permit tcp any host external_web_server_ip_address2 eq
port-number any
```

Example:

```
Device(config)# 50 permit tcp host 192.168.0.99 eq 8443 any
```

Step 10 Permit DNS TCP traffic.

Example:

```
Device(config)# access-list-number permit tcp any eq domain
```

Example:

```
Device(config)# 60 permit tcp any any eq domain
```

Step 11 Deny all the other traffic.

Example:

```
Device(config)# access-list-number deny ip any any
```

Example:

```
Device(config)# 70 deny ip any any
```

Step 12 Create the WLAN using the **wlan** *wlan-name wlan-id ssid* command.

Example:

```
Device(config)# wlan wlan-name wlan-id ssid
```

Example:

```
Device(config)# wlan EWLC3-GUEST 3 EWLC3-GUEST
```

Step 13 Configure the IPv4 WLAN web ACL

Example:

```
Device(config-wlan)# ip access-group web name
```

The variable *name* specifies the user-defined IPv4 ACL name

Step 14 Return to privileged EXEC mode.

Example:

```
Device(config-wlan)# end
```

This configuration allows EWA with multiple external web servers and ports. It supports DNS or DHCP traffic and blocks unauthorized traffic.

Configure wired guest EWA with multiple web servers, ports different than default (80/443)

Configure Wired Guest External Web Authentication (EWA) when using multiple web servers or ports other than the default ports of 80 or 443 using CLI.

Wired Guest LAN profiles do not allow manual ACL assignment directly. To support multiple web servers or custom ports, use the bypass ACL in the global parameter map.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Define an extended IPv4 access list using a name, and enter access-list configuration mode.

Example:

```
Device(config)# ip access-list extended BYPASS_ACL
```

Step 3 Allow the traffic to switch centrally.

Example:

```
Device(config)# 10 deny ip any host 192.168.0.45
```

Step 4 Allow the traffic to switch centrally

Example:

```
Device(config)# access-list-number  deny ip any host hostname
```

Example:

```
Device(config)# 20 deny ip any host 4.0.0.1
```

Step 5 Creates a parameter map and enter parameter-map webauth configuration mode.

Example:

```
Device(config)# parameter-map type webauth global
```

Step 6 Create a WebAuth bypass intercept using the ACL name.

Example:

```
Device(config-params-parameter-map)# webauth-bypass-intercept BYPASS_ACL
```

Note
You cannot manually apply an ACL to the wired guest profile to configure external web authentication with multiple IP addresses or different ports. To work around this, use the bypass ACL in the wired guest profile.

Step 7 Return to privileged EXEC mode.

Example:

```
Device(config-params-parameter-map)# end
```

The wired guest profile uses the bypass ACL, enabling external web authentication with multiple web servers or custom ports.

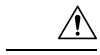
Authentication of Sleeping Client

Authenticating sleeping clients

A sleeping client is a wireless device that

- has completed web authentication and been granted guest access
- is allowed to enter sleep mode and wake up without reauthenticating through the login page, and
- the controller stores sleeping client information for a configurable duration before requiring reauthentication.

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. The sleeping client timer is activated when conditions such as idle timeout, session timeout, WLAN disabling, or AP nonoperational status occur.



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

Feature History

Feature Name	Release	Description
Webauth Sleeping Client Support	Cisco IOS XE 17.1.1s	The web authentication sleeping clients feature supports multiple combinations of authentications for a given client, which are configured on the WLAN profile.

Scenarios where sleeping clients do not need reauthentication

- 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.

Guidelines for mobility scenarios

- This feature is supported on the FlexConnect scenario with local switching and central authentication.
- L2 roaming in the same subnet is supported.
- Anchor sleeping timer is applicable.
- The sleeping client information is shared between multiple auto anchors when a sleeping client moves from one anchor to another.

Restrictions on authenticating sleeping clients

- If the MAC address of a client in sleep mode is spoofed, a fake device, such as a laptop, can be mistakenly authenticated.
- 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.
- When Layer 3 security is enabled, the Authentication, Passthrough, and On MAC Filter failure web policies are supported. The Splash Page Web Redirect web policy is 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.

Configure authentication for sleeping clients (GUI)

Enable authentication for sleeping clients to ensure devices can maintain secure network access even after entering sleep mode.

Sleeping clients, such as laptops or mobile devices, periodically enter low-power states. You configure authentication to ensure they reconnect securely and seamlessly when they wake.

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 Check **Sleeping Client Status** check box.

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

Authentication for sleeping clients is now enabled. Devices entering sleep mode will reconnect securely without manual intervention.

Configure authentication for sleeping clients (CLI)

Configure authentication parameters to manage sleeping wireless clients and control their session persistence on the device.

Sleeping clients are wireless devices that temporarily disconnect due to power-saving or roaming behaviors. Configuring authentication for sleeping clients ensures they are recognized and handled appropriately during these periods.

Procedure

Step 1 Create a parameter map and enters parameter-map webauth configuration mode.

Example:

```
Device(config)# parameter-map type webauth global
```

Step 2 Configure the sleeping client timeout to 100 minutes.

Example:

```
Device(config-params-parameter-map)# sleeping-client timeout 100
```

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 Exit parameter-map webauth configuration mode and returns to privileged EXEC mode.

Example:

```
Device# end
```

Step 4 (Optional) Show the MAC address of the clients and the time remaining in their respective sessions.

Example:

```
Device# show wireless client sleeping-client
```

Step 5 (Optional) Delete sleeping client entries from the sleeping client cache.

- **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.

Example:

```
Device# clear wireless client sleeping-client
mac-address 00e1.e1e1.0001
```

The device displays the global configuration prompt, allowing you to make configuration changes.