Web-Based Authentication

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

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

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

If authentication fails, 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, authentication forwards a Login-Expired HTML page to the host, and the user is .

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 embedded wireless controller are used during the local web authentication.
- *Customized*—The customized web pages (Login, Success, Fail, and Expire) are downloaded onto the embedded wireless 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 embedded wireless 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 embedded wireless 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:

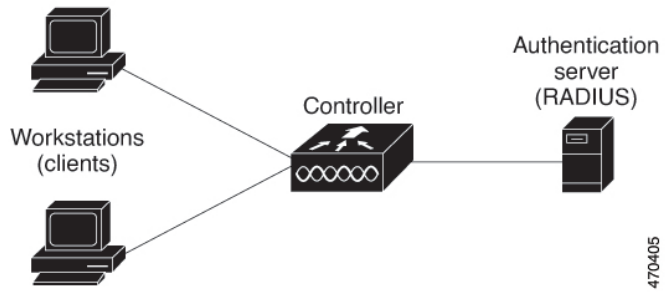
```
<body onload="loadAction();">
```

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

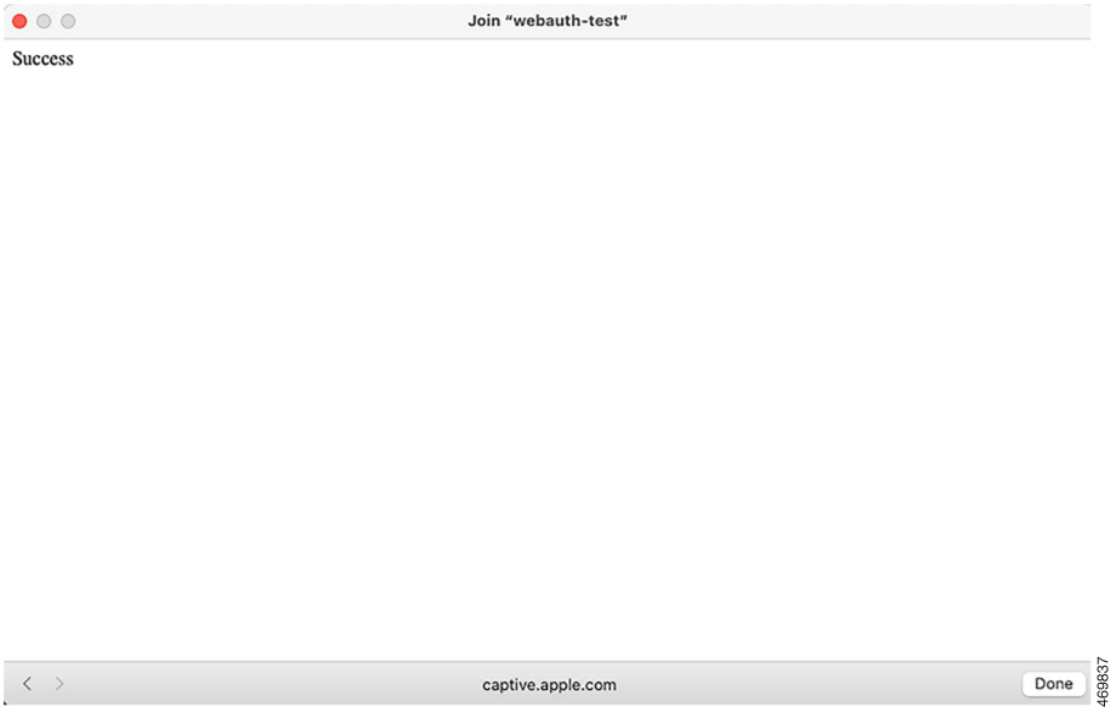
    banner text <text>

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

    parameter-map type webauth global

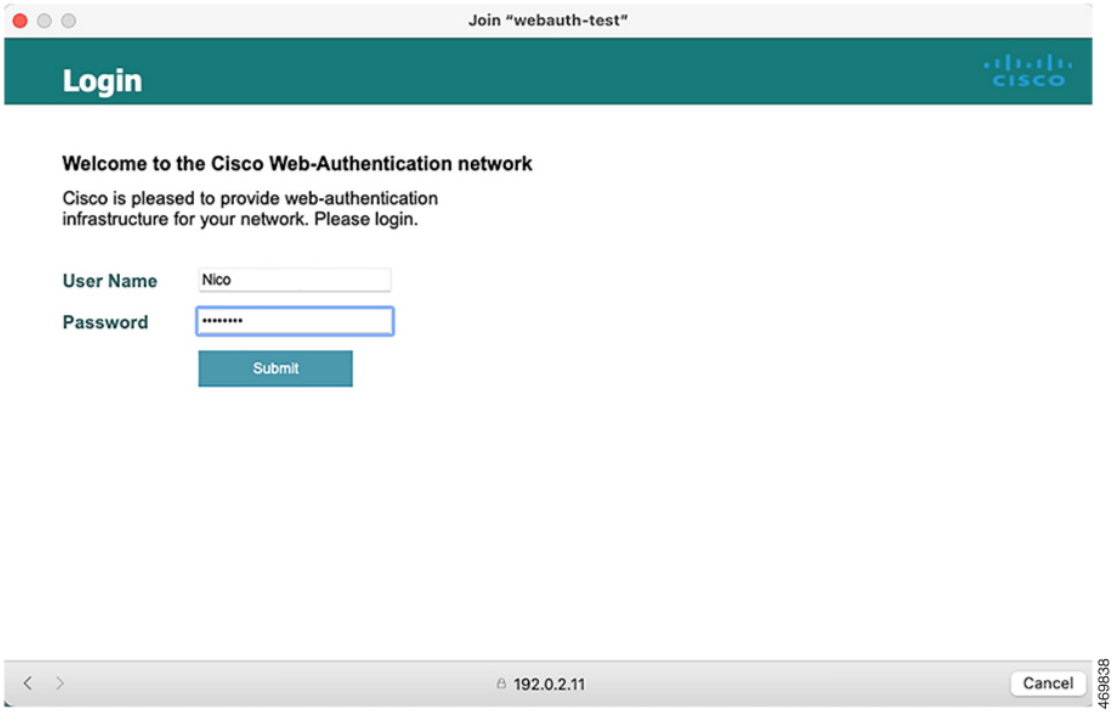
    banner file <filepath>

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.

From Cisco IOS XE Dublin 17.11.1, special characters such as *ö* or *à* are supported in the login portal for banner title and banner text. The number of characters supported on the banner text has been doubled to 400. To support special characters, ensure that you configure the **exec-character-bits** command under the line console (for serial port) or line vty (for SSH).

Note

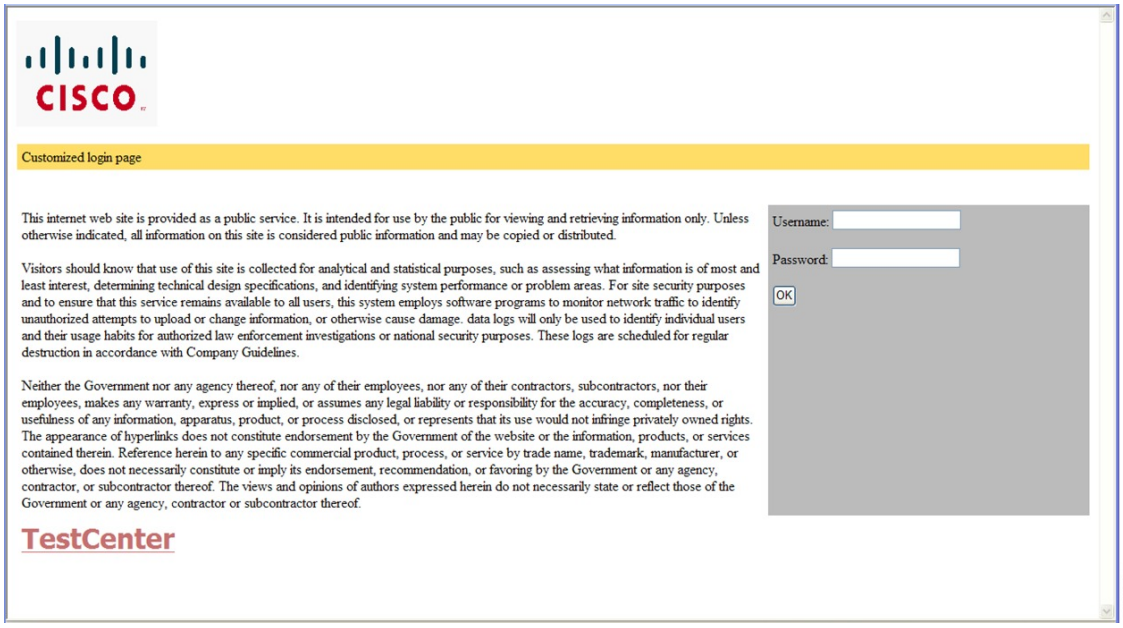
- If the banner text string exceeds the maximum limit of 400 characters, an error message is displayed and the configuration is rejected. Also, the parser has a limitation of 254 characters per line (including the CLI keywords). If you want to use more than 254 characters, ensure that you split it into two or multiple lines.
- The webauth login page displays only the default banner strings if banner command is not configured.

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

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

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:**	Enables AAA functionality.

	Command or Action	Purpose
	`Device(config)# aaa new-model`	
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 host** {*hostname* | *ip_address*} **Example:** `Device(config)# tacacs-server host 10.1.1.1`	Specifies a AAA server.

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.

Allowing Special Characters for Serial Port

Before you begin

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**line console** *line-num* **Example:** `Device(config)# line console 0`	Configures the primary terminal line number.
Step 3	**exec-timeout** *mins sec* **Example:** `Device(config-line)# exec-timeout 12 0`	Configures the time to disconnect idle EXEC sessions.
Step 4	**login authentication** *word* **default** **Example:** `Device(config-line)# login authentication NO_LOGIN`	Configures login authentication checking. It can be authentication list with a name or the default authentication list.
Step 5	**exec-character-bit** {**7** | **8**} **Example:** `Device(config-line)# exec-character-bit 8`	Configures the character widths of EXEC command characters.
Step 6	**stopbits** {**1** | **1.5** | **2**} **Example:** `Device(config-line)# stopbits 1`	Configures the stop bits for the console port.
Step 7	**end** **Example:** `Device(config-line)# end`	Returns to privileged EXEC mode.

Allowing Special Characters for VTY Port

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 global** **Example:** `Device(config)# parameter-map type webauth global`	Creates a parameter map and enters parameter-map webauth configuration mode.
Step 3	**banner text** *text* **Example:** `Device(config-params-parameter-map)# banner text #Hêllö#`	You can create a custom banner (of up to 400 characters) by entering *c* <*banner-text*> *c*, where *c* is a delimiting character. If the string exceeds the maximum limit of 400 characters, an error message is displayed and the configuration is rejected. Also, the parser has a limitation of 254 characters per line (including the CLI keywords). If you want to use more than 254 characters, ensure that you split it into two or multiple lines. The webauth login page displays only the default banner strings, if banner command is not configured.
Step 4	**end** **Example:** `Device(config-params-parameter-map)# end`	Returns to privileged EXEC mode.

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**.

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.

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.

	Command or Action	Purpose
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=[illegible]`	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:** `Device(config-params-parameter-map)# exit`	Returns to global configuration mode.
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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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.
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**	Permits the access from the external web server port 8443 to any host.

	Command or Action	Purpose
	Example: `Device(config)# 50 permit tcp host 192.168.0.99 eq 8443 any`	
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:**	Defines an extended IPv4 access list using a name, and enters access-list configuration mode.

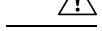
	Command or Action	Purpose
	`Device(config)# ip access-list extended BYPASS_ACL`	
Step 3	*access-list-number* **deny ip any host** *hostname* **Example:** `Device(config)# 10 deny ip any host 192.168.0.45`	Allows the traffic to switch centrally.
Step 4	*access-list-number* **deny ip any host** *hostname* **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 embedded wireless controllers in a mobility group. A client that is associated with one embedded wireless controller goes to sleep and then wakes up and gets associated with the other embedded wireless controller.
- Suppose there are three embedded wireless controllers in a mobility group. A client that is associated with the second embedded wireless controller that is anchored to the first controller goes to sleep, wakes up, and gets associated with the third embedded wireless controller.
- A client sleeps, wakes up and gets associated with the same or different export foreign embedded wireless 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.
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.

Multi Authentication Combination with 802.1X Authentication and Local Web Authentication

Feature History for Multiauthentication Combination of 802.1X and Local Web Authentication

This table provides release and related information about the feature explained in this section.

This feature is also available in all the releases subsequent to the one in which they are introduced in, unless noted otherwise.

Table 2: Feature History for Multiauthentication Combination of 802.1X and Local Web Authentication

Release	Feature	Feature Information
Cisco IOS XE Dublin 17.11.1	Multiauthentication Combination of 802.1X and Local Web Authentication	This feature supports the merging of applied policies during multiauthentication of 802.1X or MAC authentication bypass (MAB) and local web authentication (LWA).

Information About Multiauthentication Combination with 802.1X Authentication and Local Web Authentication

In a wireless setup, for example, in a university, clients authenticate through 802.1X authentication. Because the 802.1X (dot1X) authentication process is secure and does not require user intervention, the end-users are unaware of the network that their devices are connected to. This could lead to serious concerns if they connect to the university's wireless network and post inappropriate content or access restricted content.

To avoid this situation, web authentication (webauth) and 802.1X authentication are configured in the network. End-user consent is used as a part of webauth to inform users that they are connected to the university's Wi-Fi network.

When the end-users accept the credentials for consent, AAA policies are not applied. The AAA policies that were applied earlier are deleted, resulting in a VLAN change and client disconnection.

A new command is introduced in Cisco IOS XE Dublin 17.11.1 to fix this issue. When you run the **consent activation-mode merge** command, the policy that is applied through consent is merged with the policy applied for 802.1X or MAC Authentication Bypass (MAB) authentication, thereby allowing clients to access the network. This command is available in parameter-map mode, which is configured with **type consent** command.

Limitations for Multi Authentication Combination of 802.1X and Local Web Authentication

The following are the limitations for multiauthentication combination of 802.1X authentication and LWA:

- It is not possible to configure this feature on the controller GUI.

- SNMP is not supported.
- When the **consent activation-mode merge** command is not configured on the webauth parameter map, the default activation mode is Replace. This means that the user profile for consent replaces all the user profile policies that were previously applied.

Enabling the Multiauthentication Combination of 802.1X Authentication and Local Web Authentication (CLI)

Before you begin

Ensure that you have working knowledge of multiauthentication concepts, LWA (consent), and AAA override.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enter global configuration mode.
Step 2	**parameter-map type webauth** *parameter-map-name* **Example:** `Device(config)# parameter-map type webauth parameter-map1`	Configures the webauth type parameter. Enters the parameter map configuration mode.
Step 3	**type consent** **Example:** `Device(config-params-parameter-map)# type consent`	Configures the type as **consent**.
Step 4	**[no] consent** {**activation-mode merge** | **email**} **Example:** `Device(config-params-parameter-map)# consent activation-mode merge`	Enables policy activation mode and merges the previous policy. Run the **no** form of this command to disable the feature.

Verifying Multiauthentication Combination with 802.1X Authentication and Local Web Authentication

To verify the multiauthentication combination with 802.1X authentication and LWA, run the following command:

```
Device# show parameter-map type webauth lwa-consent
Parameter Map Name               : lwa_consent
  Banner Title                   : Consent Title
  Banner Text                    : Please accept the consent
  Type                           : consent
  Auth-proxy Init State time     : 300 sec
```

```
Webauth max-http connection    : 200
Webauth logout-window          : Enabled
Webauth success-window         : Enabled
Consent Email                  : Disabled
Activation Mode                : Merge
Sleeping-Client                : Disabled
Webauth login-auth-bypass:
```