ISE Simplification and Enhancements

Security utilities configuration options

You can set up the RADIUS server using this simplified method:

wireless-default radius server *ip* **key** *secret*

- Configures AAA authorization for network services, including authentication for web authentication and Dot1x.
- Enables local authentication with default authorization.
- Applies the default redirect ACL for Central Web Authentication (CWA).
- Creates a global parameter map with a virtual IP and enables captive bypass portal.
- Configures all default AAA settings required when configuring a RADIUS server.
- Sets method-list configuration and enables RADIUS accounting by default.
- Disables RADIUS aggressive failovers by default and sets the RADIUS request timeout to five seconds.
- Implements a default access control list (ACL) for CWA URL redirect.

```
aaa new-model
               aaa authentication webauth default group radius
               aaa authentication dot1x default group radius
               aaa authorization network default group radius
               aaa accounting identity default start-stop group radius
               !
               aaa server radius dynamic-author
               client <IP> server-key cisco123
               !
               radius server RAD_SRV_DEF_<IP>
               description Configured by wireless-default
               address ipv4 <IP> auth-port 1812 acct-port 1813
               key <key>
               !
```

```
aaa local authentication default authorization default
aaa session-id common
!
ip access-list extended CISCO-CWA-URL-REDIRECT-ACL-DEFAULT
remark " CWA ACL to be referenced from ISE "
deny udp any any eq domain
deny tcp any any eq domain
deny udp any eq bootps any
deny udp any any eq bootpc
deny udp any eq bootpc any
deny ip any host <IP>
permit tcp any any eq www
!
parameter-map type webauth global
captive-bypass-portal
virtual-ip ipv4 192.0.2.1
virtual-ip ipv6 2001:DB8::1
!
wireless profile policy default-policy-profile
aaa-override
local-http-profiling
local-dhcp-profiling
accounting
```

Key benefits

- Simplifies deployment of RADIUS server configuration for wireless controllers.
- Eliminates the need for step-by-step manual configuration across multiple CLI commands.
- Ensures consistent and secure default settings for AAA and authentication services.

Configure multiple RADIUS servers

To configure multiple RADIUS servers for authentication in a network.

Before you begin

Ensure you have the necessary IP addresses and shared secrets for the RADIUS servers.

Procedure

Step 1 Enter global configuration mode

Example:

```
Device# configure terminal
```

Step 2 Configure a RADIUS server

Example:

```
Device(config)# wireless-default radius server ip key secret
```

Example:

```
Device(config)# wireless-default radius server 192.2.58.90 key cisco123
```

You can configure up to ten RADIUS servers.

Step 3 Return to privileged EXEC mode

Example:

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

Alternatively, you can also press Ctrl-Z to exit global configuration mode.

Multiple RADIUS servers are now configured and ready for use in the network.

Verify AAA and Radius server configurations

To view details of AAA server, use these command:

```
Device# show run aaa
!
aaa new-model
aaa authentication webauth default group radius
aaa authentication dot1x default group radius
aaa authorization network default group radius
aaa accounting Identity default start-stop group radius
!
aaa server radius dynamic-author
 client 192.0.2.10 server-key cisco123
!
radius server RAD_SRV_DEF_192.0.2.10
 description Configured by wireless-default
 address ipv4 192.0.2.10 auth-port 1812 acct-port 1813
 key cisco123
!
aaa local authentication default authorization default
aaa session-id common
!
!
ip access-list extended CISCO-CWA-URL-REDIRECT-ACL-DEFAULT
remark " CWA ACL to be referenced from ISE "
deny udp any any eq domain
deny tcp any any eq domain
deny udp any eq bootps any
deny udp any any eq bootpc
deny udp any eq bootpc any
deny ip any host 192.0.2.10
permit tcp any any eq www
!
parameter-map type webauth global
  captive-bypass-portal
  virtual-ip ipv4 192.0.2.1
  virtual-ip ipv6 2001:DB8::1
!
wireless profile policy default-policy-profile
   aaa-override
   local-http-profiling
   local-dhcp-profiling
   accounting
```

Note The **show run aaa** output may change when new commands are added to this utility.

Captive bypassing

A captive bypass is a network behavior that

- enables client devices to automatically determine Internet connectivity and initiate authentication flows,
- may enable users to bypass captive portal splash pages through protocols such as WISPr.
- involves interaction between devices (such as Apple iOS) and Controllers, affecting web redirection and authentication.

Additional reference information

WISPr is a draft protocol that enables users to roam between different wireless service providers. Devices like Apple iOS use HTTP WISPr requests to check if they are connected to the Internet.

When direct Internet connectivity is not detected, these mechanisms prompt the device to open a web browser so the user can enter credentials for Internet access. Authentication occurs in the background whenever the device connects to a new SSID.

When an Apple iOS device connects, it sends a WISPr request to the Controller, which checks the user agent details and then triggers an HTTP request with web authentication interception. After verifying the iOS version and browser details, the Controller allows the client device to bypass captive portal settings and access the Internet.

This HTTP request triggers web authentication interception on the Controller just as any other page requests do. This leads to the standard web authentication process. If web authentication uses splash page features (such as a URL from a configured RADIUS server), the splash page might not appear for the user because WISPr requests occur at short intervals. Once a query reaches the server, web redirection or splash page display running in the background is cancelled, and the device processes the page request, which may prevent the splash page from being displayed as intended.

Example

For instance, Apple introduced an iOS feature that detects captive portals by sending a web request upon connecting to a wireless network. On iOS 6 and earlier, the request is sent to http://www.apple.com/library/test/success.html. On iOS 7 and later, multiple possible URLs are used. If a response is received, Internet access is assumed and no further interaction occurs. If no response is received, Internet access is assumed to be blocked and the device's Captive Network Assistant (CNA) auto-launches a pseudo-browser to prompt for portal login. The CNA may not function properly if it is redirected to an ISE captive portal. The Controller can suppress the pop-up of the pseudo-browser. Network administrators can configure the Controller to bypass WISPr detection. In this configuration, web authentication interception only occurs when a user intentionally requests a web page, ensuring the splash page is loaded in the user's browser context and not triggered by background connectivity probes.

Analogy: Sneaking into an amusement park

Captive bypassing is like someone sneaking into an amusement park through a side door, instead of showing their ticket at the main entrance. Some devices use automated checks to find shortcuts and access the internet without going through the intended login page, just as clever visitors might bypass security by finding an

easier way in. Network controls are needed to make sure everyone enters through the main gate and checks in properly.

Configure captive bypassing for WLAN in LWA and CWA (GUI)

Enable client devices to bypass captive portal authentication on the selected WLAN.

Before you begin

Use this task when you need certain clients to connect without passing through the captive portal, typically for pre-authorized devices.

Ensure you have administrative access to the device and the parameter map is appropriately configured.

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 **Captive Bypass Portal** check box.

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

The selected WLAN now allows captive bypass in accordance with your configuration.

Configure captive bypassing for WLAN in LWA and CWA

Enable captive bypassing for WLAN to allow users to bypass the captive portal.

This configuration is applicable in environments where users need to access the network without going through a captive portal.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Create the parameter map.

Example:

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

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

Step 3 Configure captive bypassing.

Example:

```
Device(config)# captive-bypass-portal
```

Step 4 Specify the WLAN name and ID.

Example:

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

- *profile-name* is the WLAN name which can contain 32 alphanumeric characters.
- *wlan-id* is the wireless LAN identifier. The valid range is from 1 to 512.
- *ssid-name* is the SSID which can contain 32 alphanumeric characters.

Step 5 Enable the web authentication for the WLAN.

Example:

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

Step 6 Map the parameter map.

Example:

```
Device(config-wlan)# security web-auth parameter-map parameter-map-name
```

If the parameter map is not associated with a WLAN, the configuration is considered from the global parameter map.

Step 7 (Optional) Return to privileged EXEC mode.

Example:

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

Alternatively, you can also press *Ctrl-Z* to exit global configuration mode.

The WLAN is now configured with captive bypassing enabled, allowing users to bypass the captive portal.

DHCP option 55 and 77 parameters

The DHCP sensors use these DHCP options on the ISE for both native and remote device profiling:

- Option 12: Hostname
- Option 6: Class Identifier

These options must also be sent to ISE for accurate profiling:

- Option 55: Parameter Request List
- Option 77: User Class

Configure the DHCP options 55 and 77 to ISE using the GUI

Ensure DHCP options 55 and 77 are sent to ISE to enable device profiling and policy enforcement

These options allow ISE to collect device-specific DHCP information for accurate profiling on wireless networks.

Before you begin

Confirm that your network devices are compatible with ISE profiling features.

Procedure

Step 1 Choose **Configuration** > **Tags & Profiles** > **Policy**.

Step 2 On the **Policy Profile** page, click **Add** to view the **Add Policy Profile** window.

Step 3 Click **Access Policies** tab, choose the **RADIUS Profiling** and **DHCP TLV Caching** check boxes to configure radius profiling and DHCP TLV Caching on a WLAN.

Step 4 Click **Save & Apply to Device**

he network device now sends DHCP options 55 and 77 to ISE, enabling accurate device profiling and network access policy enforcement.

Configure the device to send DHCP options 55 and 77 to ISE (CLI)

Enable the device to send specific DHCP options to ISE for improved client profiling.

This configuration is essential for environments where client profiling is necessary for network access control and monitoring.

Before you begin

Ensure that the device is running a compatible version of the software that supports DHCP options 55 and 77.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Configure WLAN policy profile and enters the wireless policy configuration mode.

Example:

```
Device(config)# wireless profile policy profile-policy
```

The *profile-policy* is the name of the WLAN policy profile being configured.

Step 3 Configure DHCP TLV caching on a WLAN.

Example:

```
Device(config-wireless-policy)# dhcp-tlv-caching
```

Enables caching of DHCP TLV information for better profiling.

Step 4 Configure client radius profiling on a WLAN.

Example:

```
Device(config-wireless-policy)# radius-profiling
```

Enables RADIUS profiling to enhance client identification.

Step 5 (Optional) Return to privileged EXEC mode.

Example:

```
Device(config-wireless-policy)# end
```

Alternatively, you can also press *Ctrl-Z* to exit global configuration mode.

The device is configured to include DHCP options 55 and 77 in client profiling information sent to Cisco ISE.

Configure the EAP request timeout using the GUI

Use the steps to configure the EAP request timeout through the GUI:

Procedure

Step 1 Choose **Configuration** > **Security** > **Advanced EAP**.

Step 2 In the **EAP-Identity-Request Timeout** field, specify the amount of time (in seconds) in which the device attempts to send an EAP identity request to wireless clients using local EAP.

Step 3 In the **EAP-Identity-Request Max Retries** field, specify the maximum number of times that the device attempts to retransmit the EAP identity request to wireless clients using local EAP.

Step 4 Set **EAP Max-Login Ignore Identity Response** to **Enabled** state to limit the number of clients that can be connected to the device with the same username. You can log in up to eight times from different clients (PDA, laptop, IP phone, and so on) on the same device. The default state is **Disabled**.

Step 5 In the **EAP-Request Timeout** field, specify the amount of time (in seconds) in which the device attempts to send an EAP request to wireless clients using local EAP.

Step 6 In the **EAP-Request Max Retries** field, specify the maximum number of times that the device attempts to retransmit the EAP request to wireless clients using local EAP.

Step 7 In the **EAPOL-Key Timeout** field, specify the amount of time (in seconds) in which the device attempts to send an EAP key over the LAN to wireless clients using local EAP.

Step 8 In the **EAPOL-Key Max Retries** field, specify the maximum number of times that the device attempts to send an EAP key over the LAN to wireless clients using local EAP.

Step 9 In the **EAP-Broadcast Key Interval** field, specify the time interval between rotations of the broadcast encryption key used for clients and click **Apply**.

Note
After configuring the EAP-Broadcast key interval to a new time period, you must shut down or restart the WLAN for the changes to take effect. Once the WLAN is shut down or restarted, the M5 and M6 packets are exchanged when the configured timer value expires.

Configure EAP request timeout

Set the timeout for EAP requests to manage client exclusion effectively.

This configuration is useful in environments where EAP authentication is used, and it is necessary to handle clients that do not respond in a timely manner.

Before you begin

Ensure you have access to the device and are in the appropriate configuration mode.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 2 Enable exclusion on timeout and no response.

Example:

```
Device(config)# wireless wps client-exclusion dot1x-timeout
```

By default, this feature is enabled. To disable, append a *no* at the beginning of the command.

Step 3 Return to privileged EXEC mode.

Example:

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

Alternatively, you can also press *Ctrl-Z* to exit global configuration mode.

The EAP request timeout is now configured, allowing for better management of client exclusions.

Configure the EAP request timeout in wireless security

Set the EAP request timeout to ensure timely authentication in wireless security.

This configuration is essential in environments where EAP is used for authentication, as it affects the responsiveness of the authentication process.

Before you begin

Ensure you have access to the device and are in privileged EXEC mode.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 2 Configure the EAP request retransmission timeout value in seconds.

Example:

```
Device(config)# wireless security dot1x request timeout timeout
```

Configures the EAP request retransmission timeout value in seconds.

Step 3 Return to privileged EXEC mode.

Example:

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

Returns to privileged EXEC mode. Alternatively, you can also press Ctrl-Z to exit global configuration mode.

The EAP request timeout is now configured, which will help in managing the authentication process in wireless security.

Captive portals

A captive portal is a network authentication mechanism that

- intercepts and redirects user web requests to a designated login or acceptance page,
- allows administrators to enforce authentication or policy acceptance before granting network access, and
- supports configuration of multiple web authentication URLs based on access point (AP), WLAN, or global scope.

Order of precedence

Captive portal configurations let you define different web authentication URLs—including external URLs—for the same SSID, based on the specific access point or WLAN. By default, the system uses a global URL for user authentication, but administrators may override this at the WLAN or AP level to customize user experience.

Order of precedence

1. AP-level configuration (highest priority)
2. WLAN-level configuration
3. Global configuration (default)

Restrictions

- This feature is supported on standalone controllers only.
- Export-Anchor configurations are not supported.

Configure captive portal (GUI)

Configure a secure WLAN that uses captive portal authentication for guest or user access.

Configure a captive portal when you require user authentication on your Wi-Fi network for guests or employees.

Before you begin

Confirm the web authentication parameter map and necessary authentication lists are already created.

Procedure

Step 1 Choose **Configuration** > **Tags & Profiles** > **WLANs**.

Step 2 Click **Add**.

Step 3 In the **General** tab, enter the **Profile Name**, the **SSID**, and the **WLAN ID**.

Step 4 In the **Security** > **Layer2** tab, uncheck the **WPA Policy**, **AES** and **802.1x** check boxes.

Step 5 In the **Security** > **Layer3** tab, choose the parameter map from the **Web Auth Parameter Map** drop-down list and authentication list from the **Authentication List** drop-down list.

Step 6 In the **Security** > **AAA** tab, choose the Authentication list from the **Authentication List** drop-down list.

Step 7 Click **Apply to Device**.

Step 8 Choose **Configuration** > **Security** > **Web Auth**.

Step 9 Choose a **Web Auth Parameter Map**.

Step 10 In the **General** tab, enter the **Maximum HTTP connections**, **Init-State Timeout(secs)** and choose **webauth** from the **Type** drop-down list.

Step 11 In the **Advanced** tab, under the **Redirect to external server** settings, enter the **Redirect for log-in** server.

Step 12 Click **Update & Apply**.

The captive portal WLAN is available, and users associating with this SSID will be redirected for authentication as defined in the parameter map.

Configure captive portal

Set up a captive portal to manage user access to the network.

The captive portal is used to authenticate users before granting them access to the network, typically in public Wi-Fi environments.

Before you begin

Ensure that the WLAN is properly configured and that you have the necessary access rights.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 2 Configure the WLAN profile.

Example:

```
Device(config)# wlan profile-name
```

Enables or disables all WLANs and creates the WLAN identifier. The *profile-name* and the SSID network name should be up to 32 alphanumeric characters.

Step 3 Configure the WLAN web ACL.

Example:

```
Device(config-wlan)# ip access-group web IPv4-ACL-Name
```

WLAN needs to be disabled before performing this operation.

Step 4 Disable WPA security.

Example:

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

Step 5 Disable security AKM for dot1x.

Example:

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

Disables security AKM for dot1x.

Step 6 Disable WPA2 ciphers for AES.

Example:

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

Disables WPA2 ciphers for AES.

Step 7 Enable web authentication for WLAN.

Example:

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

Enables web authentication for WLAN. Here,

- **authentication-list**

  authentication-list-name : Sets the authentication list for IEEE 802.1x.

- **authorization-list**

  authorization-list-name : Sets the override-authorization list for IEEE 802.1x.

- **on-macfilter-failure** : Enables Web authentication on MAC filter failure.

- **parameter-map**

  parameter-map-name : Configures the parameter map.

Note
When **security web-auth** is enabled, you get to map the default **authentication-list** and global **parameter-map** . This is applicable for authentication-list and parameter-map that are not explicitly mentioned.

Step 8 (Optional) Enable the WLAN.

Example:

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

Step 9 Exit from the WLAN configuration.

Example:

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

Step 10 Create a parameter map and enter parameter-map webauth configuration mode.

Example:

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

Creates a parameter map and enters parameter-map webauth configuration mode.

Step 11 Configure the webauth type parameter.

Example:

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

Step 12 Configure the WEBAUTH timeout in seconds.

Example:

```
Device(config-params-parameter-map)# timeout init-state sec <timeout-seconds>
```

Valid range for the time in sec parameter is 60 seconds to 3932100 seconds.

Step 13 Configure the URL string for redirect during login.

Example:

```
Device(config-params-parameter-map)# redirect for-login <URL-String>
```

Example:

```
Device(config-params-parameter-map)# redirect for-login
https://172.16.100.157/portal/login.html
```

Configures the URL string for redirect during login.

Step 14 Exit the parameters configuration.

Example:

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

Step 15 Configure policy tag and enter policy tag configuration mode.

Example:

```
Device(config)# wireless tag policy policy-tag-name
```

Step 16 Attach a policy profile to a WLAN profile.

Example:

```
Device(config-policy-tag)# wlan wlan-profile-name policy policy-profile-name
```

Step 17 Save the configuration and exit configuration mode.

Example:

```
Device(config-policy-tag)# end
```

The captive portal is now configured and ready to manage user access to the network.

Example: Captive portal configuration

The example shows how you can have APs at different locations, broadcasting the same SSID but redirecting clients to different redirect portals:

Configuring multiple parameter maps pointing to different redirect portal:

```
parameter-map type webauth parMap1
type webauth
timeout init-state sec 21600
redirect for-login
https://172.16.12.3:8080/portal/PortalSetup.action?portal=cfdbce00-2ce2-11e8-b83c-005056a06b27
redirect portal ipv4 172.16.12.3
!
!
parameter-map type webauth parMap11
type webauth
timeout init-state sec 21600
redirect for-login
https://172.16.12.4:8443/portal/PortalSetup.action?portal=094e7270-3808-11e8-9797-02421e4cae0c
redirect portal ipv4 172.16.12.4
!
```

Associating these parameter maps to different WLANs:

```
wlan edc1 1 edc
ip access-group web CPWebauth
no security wpa
no security wpa akm dot1x
no security wpa wpa2 ciphers aes
security web-auth
security web-auth authentication-list cp-webauth
security web-auth parameter-map parMap11
no shutdown
wlan edc2 2 edc
ip access-group web CPWebauth
no security wpa
no security wpa akm dot1x
no security wpa wpa2 ciphers aes
security web-auth
security web-auth authentication-list cp-webauth
security web-auth parameter-map parMap1
no shutdown
```

Note All WLANs have identical SSIDs.

Associating WLANs to different policy tags:

```
wireless tag policy policy_tag_edc1
wlan edc1 policy policy_profile_flex
wireless tag policy policy_tag_edc2
wlan edc2 policy policy_profile_flex
```

Assigning these policy tags to the desired APs:

```
ap E4AA.5D13.14DC
policy-tag policy_tag_edc1
site-tag site_tag_flex
ap E4AA.5D2C.3CAC
policy-tag policy_tag_edc2
site-tag site_tag_flex
```