ISE Simplification and Enhancements

Utilities for Configuring Security

This chapter describes how to configure all the RADIUS server side configuration using the following command:

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

This simplified configuration option provides the following:

- Configures AAA authorization for network services, authentication for web auth and Dot1x.

- Enables local authentication with default authorization.

- Configures the default redirect ACL for CWA.

- Creates global parameter map with virtual IP and enables captive bypass portal.

- Configures all the AAA configuration for a default case while configuring the RADIUS server.

- The method-list configuration is assumed by default on the WLAN.

- Enables the radius accounting by default.

- Disables the radius aggressive failovers by default.

- Sets the radius request timeouts to 5 seconds by default.

- Enables captive bypass portal.

This command configures the following in the background:

```
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 1001::1
!
wireless profile policy default-policy-profile
  aaa-override
  local-http-profiling
  local-dhcp-profiling
  accounting
```

Thus, you need not go through the entire Configuration Guide to configure wireless embedded wireless controller for a simple configuration requirement.

Configuring Multiple Radius Servers

Use the following procedure to configure a RADIUS server.

Procedure

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

Step 2	**wireless-default radius server** *ip* **key** *secret*
	
	Example:
	
	`Device(config)# wireless-default radius`
	`server 9.2.58.90 key cisco123`
		Configures a radius server.
		
		Note You can configure up to ten RADIUS servers.

Step 3	**end** **Example:** `Device(config)#` **end**	Returns to privileged EXEC mode. Alternatively, you can also press **Ctrl-Z** to exit global configuration mode.

Verifying AAA and Radius Server Configurations

To view details of AAA server, use the following 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 9.2.58.90 server-key cisco123
!
radius server RAD_SRV_DEF_9.2.58.90
 description Configured by wireless-default
 address ipv4 9.2.58.90 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 9.2.58.90
permit tcp any any eq www
!
parameter-map type webauth global
  captive-bypass-portal
  virtual-ip ipv4 192.0.2.1
  virtual-ip ipv6 1001::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.

Configuring Captive Portal Bypassing for Local and Central Web Authentication

Information About Captive Bypassing

WISPr is a draft protocol that enables users to roam between different wireless service providers. Some devices (For example, Apple iOS devices) have a mechanism using which they can determine if the device is connected

to Internet, based on an HTTP WISPr request made to a designated URL. This mechanism is used for the device to automatically open a web browser when a direct connection to the internet is not possible. This enables the user to provide his credentials to access the internet. The actual authentication is done in the background every time the device connects to a new SSID.

The client device (Apple iOS device) sends a WISPr request to the embedded wireless controller, which checks for the user agent details and then triggers an HTTP request with a web authentication interception in the embedded wireless controller. After verification of the iOS version and the browser details provided by the user agent, the embedded wireless controller allows the client to bypass the captive portal settings and provides access to the Internet.

This HTTP request triggers a web authentication interception in the embedded wireless controller as any other page requests are performed by a wireless client. This interception leads to a web authentication process, which will be completed normally. If the web authentication is being used with any of the embedded wireless controller splash page features (URL provided by a configured RADIUS server), the splash page may never be displayed because the WISPr requests are made at very short intervals, and as soon as one of the queries is able to reach the designated server, any web redirection or splash page display process that is performed in the background is cancelled, and the device processes the page request, thus breaking the splash page functionality.

For example, Apple introduced an iOS feature to facilitate network access when captive portals are present. This feature detects the presence of a captive portal by sending a web request on connecting to a wireless network. This request is directed to http://www.apple.com/library/test/success.html for Apple iOS version 6 and older, and to several possible target URLs for Apple iOS version 7 and later. If a response is received, then the Internet access is assumed to be available and no further interaction is required. If no response is received, then the Internet access is assumed to be blocked by the captive portal and Apple's Captive Network Assistant (CNA) auto-launches the pseudo-browser to request portal login in a controlled window. The CNA may break when redirecting to an ISE captive portal. The embedded wireless controller prevents this pseudo-browser from popping up.

You can now configure the embedded wireless controller to bypass WISPr detection process, so the web authentication interception is only done when a user requests a web page leading to splash page load in user context, without the WISPr detection being performed in the background.

Configuring Captive Bypassing for WLAN in LWA and CWA (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 **Captive Bypass Portal** check box.

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

Configuring Captive Bypassing for WLAN in LWA and CWA (CLI)

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**parameter-map type webauth** *parameter-map-name* **Example:** `Device(config)# parameter-map type webauth WLAN1_MAP`	Creates the parameter map. The *parameter-map-name* must not exceed 99 characters.
Step 3	**captive-bypass-portal** **Example:** `Device(config)# captive-bypass-portal`	Configures captive bypassing.
Step 4	**wlan** *profile-name wlan-id ssid-name* **Example:** `Device(config)# wlan WLAN1_NAME 4 WLAN1_NAME`	Specifies the WLAN name and ID. • *profile-name* is the WLAN name which can contain 32 alphanumeric characters. • *wlan-id* is the wireless LAN identifier. The valid range is from 1 to 512. • *ssid-name* is the SSID which can contain 32 alphanumeric characters.
Step 5	**security web-auth** **Example:** `Device(config-wlan)# security web-auth`	Enables the web authentication for the WLAN.
Step 6	**security web-auth parameter-map** *parameter-map-name* **Example:** `Device(config-wlan)# security web-auth parameter-map WLAN1_MAP`	Maps the parameter map. **Note** If parameter map is not associated with a WLAN, the configuration is considered from the global parameter map.
Step 7	**end** **Example:** `Device(config-wlan)# end`	Returns to privileged EXEC mode. Alternatively, you can also press **Ctrl-Z** to exit global configuration mode.

Sending DHCP Options 55 and 77 to ISE

Information about DHCP Option 55 and 77

The DHCP sensors use the following DHCP options on the ISE for native and remote profiling:

- **Option 12**: Hostname
- **Option 6**: Class Identifier

Along with this, the following options needs to be sent to the ISE for profiling:

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

Configuration to Send DHCP Options 55 and 77 to ISE (GUI)

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**.

Configuration to Send DHCP Options 55 and 77 to ISE (CLI)

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**wireless profile policy** *profile-policy* **Example:** `Device(config)# wireless profile policy rr-xyz-policy-1`	Configures WLAN policy profile and enters the wireless policy configuration mode.
Step 3	**dhcp-tlv-caching** **Example:**	Configures DHCP TLV caching on a WLAN.

	Command or Action	Purpose
	`Device(config-wireless-policy)# dhcp-tlv-caching`	
Step 4	**radius-profiling** **Example:** `Device(config-wireless-policy)# radius-profiling`	Configures client radius profiling on a WLAN.
Step 5	**end** **Example:** `Device(config-wireless-policy)# end`	Returns to privileged EXEC mode. Alternatively, you can also press **Ctrl-Z** to exit global configuration mode.

Configuring EAP Request Timeout (GUI)

Follow the steps given below 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.

Configuring EAP Request Timeout

Procedure

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

Step 2	**wireless wps client-exclusion dot1x-timeout**
	
	Example:
	`Device(config)# wireless wps`
	`client-exclusion dot1x-timeout`
		Enables exclusion on timeout and no response.
		
		By default, this feature is enabled.
		
		To disable, append a **no** at the beginning of the command.

Step 3	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode. Alternatively, you can also press **Ctrl-Z** to exit global configuration mode.

Configuring EAP Request Timeout in Wireless Security (CLI)

Procedure

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

Step 2	**wireless security dot1x request** {**retries** *0 - 20* | **timeout** *1 - 120*}
	
	Example:
	`Device(config)# wireless security dot1x`
	`request timeout 60`
		Configures the EAP request retransmission timeout value in seconds.

Step 3	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode. Alternatively, you can also press Ctrl-Z to exit global configuration mode.

Captive Portal

Captive Portal Configuration

This feature enables you to configure multiple web authentication URLs (including external captive URLs) for the same SSID based on an AP. The default setting is to use the Global URL for authentication. The override option is available at WLAN and AP level.

The order of precedence is:

- AP

- WLAN

- Global configuration

Restrictions for Captive Portal Configuration

- This configuration is supported in a standalone controller only.

- Export-Anchor configuration is not supported.

Configuring Captive Portal (GUI)

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**.

Configuring Captive Portal

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**wlan** {*profile-name* | **shutdown**} *network-name* **Example:** `Device(config)# wlan edc6 6 edc`	Configures the WLAN profile. Enables or Disables all WLANs and creates the WLAN identifier. The profile-name and the SSID network name should be up to 32 alphanumeric charcters.
Step 3	**ip** {**access-group** | **verify**} **web** *IPv4-ACL-Name* **Example:** `Device(config-wlan)# ip access-group web CPWebauth`	Configures the WLAN web ACL. **Note** WLAN needs to be disabled before performing this operation.
Step 4	**no security wpa** **Example:** `Device(config-wlan)# no security wpa`	Disables WPA security.
Step 5	**no security wpa akm dot1x** **Example:** `Device(config-wlan)# no security wpa akm dot1x`	Disables security AKM for dot1x.
Step 6	**no security wpa wpa2 ciphers aes** **Example:** `Device(config-wlan)# no security wpa wpa2 ciphers aes`	Disables WPA2 ciphers for AES.

Step 7	**security web-auth** {**authentication-list** *authentication-list-name* | **authorization-list** *authorization-list-name* | **on-macfilter-failure** | **parameter-map** *parameter-map-name*}
	
	Example:
	
	`Device(config-wlan)# security web-auth authentication-list cp-webauth`
	
	`Device(config-wlan)# security web-auth parameter-map parMap6`
		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.

	Command or Action	Purpose
	• **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	**no shutdown** **Example:** `Device(config-wlan)# no shutdown`	Enables the WLAN.
Step 9	**exit** **Example:** `Device(config-wlan)# exit`	Exits from the WLAN configuration.
Step 10	**parameter-map type webauth** *parameter-map-name* **Example:** `Device(config)# parameter-map type webauth parMap6`	Creates a parameter map and enters parameter-map webauth configuration mode.
Step 11	**parameter-map type webauth** *parameter-map-name* **Example:** `Device(config)# parameter-map type webauth parMap6`	Creates a parameter map and enters parameter-map webauth configuration mode.
Step 12	**type webauth** **Example:** `Device(config-params-parameter-map)# type webauth`	Configures the webauth type parameter.
Step 13	**timeout init-state sec** *<timeout-seconds>* **Example:** `Device(config-params-parameter-map)# timeout inti-state sec 3600`	Configures the WEBAUTH timeout in seconds. Valid range for the time in sec parameter is 60 seconds to 3932100 seconds.
Step 14	**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.

	Command or Action	Purpose

Step 15	**exit**
	
	Example:
	`Device(config-params-parameter-map)#`
	`exit`
		Exits the parameters configuration.

Step 16	**wireless tag policy** *policy-tag-name*
	
	Example:
	`Device(config)# wireless tag policy`
	`policy_tag_edc6`
		Configures policy tag and enters policy tag configuration mode.

Step 17	**wlan** *wlan-profile-name* **policy** *policy-profile-name*
	
	Example:
	`Device(config-policy-tag)# wlan edc6`
	`policy policy_profile_flex`
		Attaches a policy profile to a WLAN profile.

Step 18	**end** **Example:** `Device(config-policy-tag)# end`	Saves the configuration and exits configuration mode and returns to privileged EXEC mode.

Captive Portal Configuration - Example

The following 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
```

Captive Portal Configuration - Example