

Cisco Industrial Wireless Workgroup Bridge and Universal WGB Deployment Guide

First Published: 2022-05-17

Last Modified: 2023-05-11

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2022 Cisco Systems, Inc. All rights reserved.



CONTENTS

Full Cisco Trademarks with Software License ?

CHAPTER 1 **Introduction** 1

- Overview of Workgroup Bridge 1
- Overview of Universal WGB 2
- Supported Platforms 2
  - Cisco Catalyst IW6300 Heavy Duty Series and 6300 Series Embedded Services Access Points 3
  - Cisco Wide Pluggable Form Factor WIFI6 AP Module 3
- Limitations and Restrictions 4

CHAPTER 2 **Configuring uWGB** 5

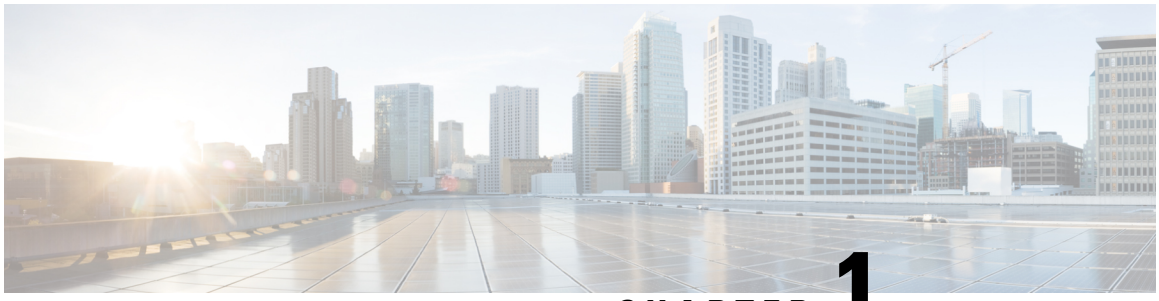
- Configuring AP to uWGB Mode 5
- Configure IP address 6
  - Configure IPv4 address 6
  - Configure IPv6 address 6
- Configure a Dot1X credential 7
- Configure an EAP profile 7
- Configure trustpoint manual enrollment for terminal 8
- Configure trustpoint auto-enrollment for WGB 9
- Configure manual certificate enrollment using TFTP server 10
- SSID configuration 11
  - Create an SSID profile 11
    - Configure an SSID profile using open authentication 11
    - Configure an SSID profile using PSK authentication 12
    - Configure an SSID profile using Dot1x authentication 12
  - Configuring Radio Interface for uWGB 13

Configuring Workgroup Bridge Timeouts 13
Flex Antenna Band Configuration 14

CHAPTER 3 Configuring WGB 15
Configuring AP to WGB Mode 15
Configure IP address 16
Configure IPv4 address 16
Configure IPv6 address 16
Configure a Dot1X credential 17
Deauthenticate WGB wired client 17
Configure an EAP profile 17
Configure trustpoint manual enrollment for terminal 18
Configure trustpoint auto-enrollment for WGB 19
Configure manual certificate enrollment using TFTP server 20
SSID configuration 21
Create an SSID profile 21
Configure an SSID profile using open authentication 22
Configure an SSID profile using PSK authentication 22
Configure an SSID profile using Dot1x authentication 22
Configuring Radio Interface for Workgroup Bridges 23
Configuring Workgroup Bridge Timeouts 24
Flex Antenna Band Configuration 24

CHAPTER 4 Workgroup Bridge FAQ 25
Workgroup Bridge FAQ 25

CHAPTER 5 Troubleshooting 27
Debug Commands 27
WGB Show Commands 27
uWGB Show Commands 28
WGB Debug Examples 29



CHAPTER 1

Introduction

This document provides information about Workgroup Bridge (WGB) and Universal WGB mode that are supported on the Cisco Industrial Wireless Cheetah OS (COS) based access points.

Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

- Overview of Workgroup Bridge, on page 1
- Overview of Universal WGB, on page 2
- Supported Platforms, on page 2
- Limitations and Restrictions, on page 4

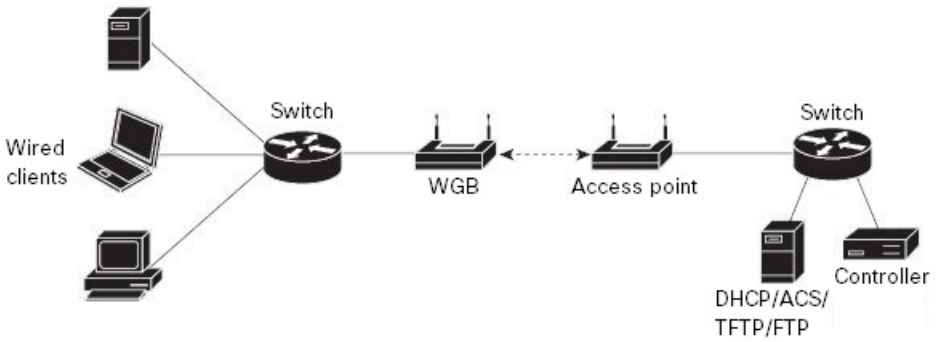
Overview of Workgroup Bridge

A workgroup bridge (WGB) is a Cisco access point that can be configured in a mode that permits it to associate with a wireless infrastructure, providing network access on behalf of wired clients. It is also called as a wireless bridge.

Note This document only covers WGB mode on the Cheetah OS (COS) APs.

A Cisco WGB provides information about its wired clients via Internet Access Point Protocol (IAPP) messaging. This enables the wireless infrastructure to know the MAC addresses of the WGB's wired clients. Up to 20 wired clients are supported behind a Cisco WGB.

Figure 1: WGB Example



- The following authentication modes are supported for use with a WGB:

  Open, PSK, dot1X (including LEAP, PEAP, FAST, TLS)

- Wired clients connected to the WGB are not authenticated for security. Instead, the WGB is authenticated against the access point to which it associates. Therefore, we recommend that you physically secure the wired side of the WGB.
- Wired clients connected to a WGB inherit the WGB's QoS and AAA override attributes.
- To enable the WGB to communicate with the lightweight access point, create a WLAN and make sure that Aironet IE is enabled. To associate to a Cisco AP, make sure that Aironet IE is enabled on the controller.

Overview of Universal WGB

Universal WGB (uWGB) is a complementary mode of WGB feature that acts as a wireless bridge between the wired client connected to uWGB and wireless infrastructure including Cisco and non-Cisco wireless network. One of the wireless interface is used to connect with the access point. The radio MAC is used to associate AP.

The uWGB mode only supports bridge assigned MAC address wired client to AP or Controller network. When the WGB device is in uWGB mode, only one wired client can be connected behind it. The uWGB mode does not support multiple VLANs.

Supported Platforms

The WGB and uWGB configurations discussed in this document are supported on the following Cheetah (COS) based access points:

- Cisco Catalyst IW6300 Heavy Duty Series and 6300 Series Embedded Services Access Points
- Cisco Wide Pluggable Form Factor WIFI6 AP Module

Cisco Catalyst IW6300 Heavy Duty Series and 6300 Series Embedded Services Access Points

Designed for the most hazardous industrial locations, Cisco Catalyst IW6300 Heavy Duty Access Points (hereafter called *IW6300*) deliver wireless connectivity, IoT control, and robust data collection to dangerous environments. With 802.11ac Wave 2 connectivity, dual Power over Ethernet Plus (PoE+) out for IoT sensors or peripherals, multiple power-in sources, and a variety of uplink options, the IW6300 is a flexible wireless solution today's dynamic industry landscape requires.

Cisco 6300 Series Embedded Services Access Points (hereafter called *ESW6300*) integrate wireless mesh networking into heavy-industry and smart-city assets, and provides a dependable and secure connectivity solution in almost any work environment.

The IW6300 and ESW6300 access points can operate in the following modes:

- Unified mode
  - Local
  - Flexconnect
  - Bridge
  - Flexconnect with Bridge
  - Sniffer
- Workgroup Bridge

This document covers only Workgroup Bridge (WGB) configuration.

For more information about IW6300 and ESW6300 access points, see https://www.cisco.com/c/en/us/support/wireless/industrial-wireless-6300-series/series.html.

Cisco Wide Pluggable Form Factor WIFI6 AP Module

The Cisco Wide Pluggable Form Factor WIFI6 AP Module (Cisco PID: WP-WIFI6-x) is a pluggable 802.11ax module for industrial routers. This ruggedized wide pluggable module provides the latest Wi-Fi technology and is compatible with the latest wireless controllers from Cisco. The module can run in Control and Provisioning of Wireless Access Points (CAPWAP) mode and Embedded Wireless Controller (EWC) mode, as well as Work Group Bridge (WGB) mode.

This document covers only WGB and uWGB mode configurations.



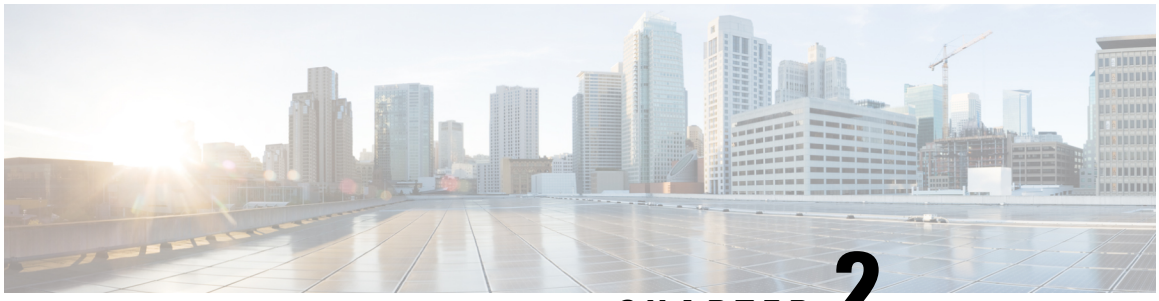
Note WP-WiFi6 supports uWGB mode from Cisco IOS XE Release 17.8.1.

For more information on configuring this module, see https://www.cisco.com/c/en/us/td/docs/routers/access/IR1800/software/b-cisco-ir1800-scg/m-wifi.html.

Limitations and Restrictions

This section provides limitations and restrictions for WGB and uWGB modes.

- The WGB can associate only with Cisco lightweight access points. The universal WGB can associate to a third party WGB.
- Per-VLAN Spanning Tree (PVST) and packets are used to detect and prevent loops in the wired and wireless switching networks. WGB transparently bridge STP packets. WGB can bridge STP packets between two wired segments. Incorrect or inconsistent configuration of STP in the wired segments can cause WGB wireless link to be blocked by the connected switch(es) to Access Point or WGB. This could cause WGB to disconnect from AP or AP disconnection to Controller to drop, and wired clients not receiving IP addresses, as STP begins to block switch port in the wired network. If administrator needs to disable bridging of STP between the wired segments by the WGB, we recommend disabling the STP on the directly connected switches in the wireless network.
- The following features are not supported for use with a WGB:
  - Idle timeout
  - Web authentication
- With Layer 3 roaming, if you plug a wired client into the WGB network after the WGB has roamed to another controller (for example, to a foreign controller), the wired client's IP address displays only on the anchor controller, not on the foreign controller.
- When you deauthenticate a WGB record from a controller, all of the WGB wired clients' entries are also deleted.
- These features are not supported for wired clients connected to a WGB:
  - MAC filtering
  - Link tests
  - Idle timeout
- Associating a WGB to a WLAN that is configured for Adaptive 802.11r is not supported.
- PoE Out is not supported for WGB mode on IW6300 and ESW6300 access points.
- WGB supports IPv6 only when IPv4 is enable. But there is no impact on WGB wired clients IPv6 traffic.
- WGB management IPv6 does not work after WGB uplink association is completed. WGB can get an IPv6 address when the association is successful. But IPv6 ping will not be passed from or to WGB. SSH from wireless or wired client to WGB management IPv6 is not working. The workaround to bypass the pingable issue is to re-enable IPv6, even though IPv6 has already been enabled and the IPv6 address has been assigned.
- uWGB mode does not support SSH connecting to itself.
- uWGB mode does not support TFTP or SFTP upgrade image. The workaround is to convert uWGB mode to CAPWAP AP or WGB mode connected with Cisco AP to upgrade the image.
- uWGB does not support host IP service. Some functions, such as image upgrade via radio uplink and remote management via SSH session, are not supported.



CHAPTER 2

Configuring uWGB

This chapter contains these topics:

- Configuring AP to uWGB Mode, on page 5
- Configure IP address, on page 6
- Configure a Dot1X credential, on page 7
- Configure an EAP profile, on page 7
- Configure trustpoint manual enrollment for terminal, on page 8
- Configure trustpoint auto-enrollment for WGB , on page 9
- Configure manual certificate enrollment using TFTP server, on page 10
- SSID configuration, on page 11
- Configuring Workgroup Bridge Timeouts, on page 13
- Flex Antenna Band Configuration, on page 14

Configuring AP to uWGB Mode

Cisco 802.11ac wave2 AP (IW6300 and ESW6300) and 802.11ax AP module (WP-WIFI6) are Cheetah OS (COS) based access points. The COS WGB function runs on the following image versions:

- **ap3g3-k9w8-tar.xxx.tar**
- **ap1g8-k9w8-tar.xxx.tar**

Make sure that you use the correct image version for WGB deployment.

Follow these steps to configure a Cisco AP from CAPWAP mode to uWGB mode:

1. Convert CAPWAP AP to WGB mode.

```
Wgb# ap-type workgroup-bridge
WGB is a wireless client that serve as nonroot ap for wired clients.
AP is the Master/CAPWAP AP, system will need a reboot when ap type is
changed to WGB. Do you want to proceed? (y/N):y
```

2. Configure SSID profile.

```
Wgb# configure ssid-profile <SSID_profile_name> ssid <SSID_name> authentication open
```

3. Configure radio interface to uWGB mode and map the SSID profile.

```
Wgb# configure dot11 <0/1 radio interface> mode uwgb <uwgb_wired_client_mac_address>
ssid-profile <ssid-profile>
```

Configure IP address

Configure IPv4 address

- Use the **configure ap address** *ipv4 dhcp* command to configure IPv4 address using DHCP.

```
Device#configure ap address ipv4 dhcp
```

- Use the **configure ap address** *ipv4 static ipv4_addr netmask gateway* command to configure the static IPv4 address. By doing so, you can manage the device using a wired interface without an uplink connection.

```
Device#configure ap address ipv4 static ipv4_addr netmask gateway
```

Verify current IP configuration

Use **show ip interface brief** command to view the current IP address configuration.

```
Device#show ip interface brief
```

Configure IPv6 address

Use the **configure ap address ipv6 static** *ipv6_addr prefixlen [gateway]* command to configure the static IPv6 address. This configuration allows you to manage the AP through a wired interface without uplink connection.

```
Device#configure ap address ipv6 static ipv6_addr prefixlen [gateway]
```

Enable IPv6 auto configuration

Use the **configure ap address ipv6 auto-config enable** command to enable the IPv6 auto configuration on the AP.

```
Device#configure ap address ipv6 auto-config enable
```

Note

- Use the **configure ap address ipv6 auto-config disable** command to disable the IPv6 auto configuration on the AP.
- Use the **configure ap address ipv6 auto-config enable** command to enable IPv6 SLAAC. Note that SLAAC does not apply to CoS of WGB. This command configures IPv6 address with DHCPv6 instead of SLAAC.

Configure IPv6 address using DHCP

Use the **configure ap address** *ipv6 dhcp* command to configure IPv6 address using DHCP.

```
Device#configure ap address ipv6 dhcp
```

Verify current IP configuration

Use the **show ipv6 interface brief** command to verify current IP address configuration.

```
Device#show ipv6 interface brief
```

Configure a Dot1X credential

Use the **configure dot1x credential** *profile-name* **username** *name* **password** *pwd* command to configure Dot1x credential.

```
Device#configure dot1x credential profile-name username name password pwd
```

Verify WGB EAP Dot1x profile

Use the **show wgb eap dot1x credential profile** command to view the status of WGB EAP Dot1x profile.

```
Device#show wgb eap dot1x credential profile
```

Configure an EAP profile

Perform these steps to configure an EAP profile:

1. Attach the Dot1x credential profile to the EAP profile.
2. Attach the EAP profile to the SSID profile.
3. Attach the SSID profile to the radio.

Procedure

Step 1 Use the **configure eap-profile** *profile-name* **method** {**fast** | **leap** | **peap** | **tls**} command to configure the EAP profile.

```
Device#configure eap-profile profile-name method { fast | leap | peap | tls}
```

Note
Choose an EAP profile method.

- fast
- peap, or
- tls.

Step 2 Use the **configure eap-profile** *profile-name* **trustpoint** {**default** | **name** *trustpoint-name*} command to attach the CA trustpoint for TLS. By default, the WGB uses the internal MIC certificate for authentication.

```
Device#configure eap-profile profile-name trustpoint { default | name trustpoint-name}
```

Step 3 Use the **configure eap-profile** *profile-name* **dot1x-credential** *profile-name* command to attach the dot1x-credential profile.

```
Device#configure eap-profile profile-name dot1x-credential profile-name
```

Step 4 [Optional] Use the **configure eap-profile** *profile-name* **delete** command to delete an EAP profile.

```
Device#configure eap-profile profile-name delete
```

Configure trustpoint manual enrollment for terminal

Procedure

Step 1 Use the **configure crypto pki trustpoint** *ca-server-name* **enrollment terminal** command to create a trustpoint in WGB.

```
Device#configure crypto pki trustpoint ca-server-name enrollment terminal
```

Step 2 Use the **configure crypto pki trustpoint** *ca-server-name* **authenticate** command to authenticate a trustpoint manually.

```
Device#configure crypto pki trustpoint ca-server-name authenticate
```

Enter the base 64 encoded CA certificate.

Enter **quit** to finish the certificate.

Note
If you use an intermediate certificate, import all the certificate chains in the trustpoint.

Example:

```
Device#configure crypto pki trustpoint demotp authenticate

Enter the base 64 encoded CA certificate.
....And end with the word "quit" on a line by itself....

-----BEGIN CERTIFICATE-----
[base64 encoded root CA certificate]
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
[base64 encoded intermediate CA certificate]
-----END CERTIFICATE-----
quit
```

Step 3 Use the **configure crypto pki trustpoint** *ca-server-name* **key-size** *key-length* command to configure a private key size.

```
Device#configure crypto pki trustpoint ca-server-name key-size key-length
```

Step 4 Use the **configure crypto pki trustpoint** *ca-server-name* **subject-name** *name* `[Optional]` *2ltr-country-code state-name locality org-name org-unit email* command to configure the subject-name.

```
Device#configure crypto pki trustpoint ca-server-name subject-name name [Optional] 2ltr-country-code
 state-name locality org-name org-unit email
```

Step 5 Use the **configure crypto pki trustpoint** *ca-server-name* **enroll** command to generate a private key and certificate signing request (CSR).

```
Device#configure crypto pki trustpoint ca-server-name enroll
```

Create the digitally signed certificate using the CSR output in the CA server.

Step 6 Use the **configure crypto pki trustpoint** *ca-server-name* **import certificate** command to import the signed certificate in WGB.

```
Device#configure crypto pki trustpoint ca-server-name import certificate
```

Enter the base 64 encoded CA certificate.

Enter **quit** to finish the certificate.

```
Device#quit
```

Step 7 [Optional] Use the **configure crypto pki trustpoint** *trustpoint-name* **delete** command to delete a trustpoint.

```
Device#configure crypto pki trustpoint trustpoint-name delete
```

Step 8 Use the **show crypto pki trustpoint** command to view the trustpoint summary.

```
Device#show crypto pki trustpoint
```

Step 9 Use the **show crypto pki trustpoint** *trustpoint-name* **certificate** command to view the content of the certificates that are created for a trustpoint.

```
Device#show crypto pki trustpoint trustpoint-name certificate
```

Configure trustpoint auto-enrollment for WGB

Procedure

Step 1 Use the **configure crypto pki trustpoint** *ca-server-name* **enrollment url** *ca-server-url* command to enroll a trustpoint in the WGB using the server URL.

```
Device#configure crypto pki trustpoint ca-server-name enrollment url ca-server-url
```

Step 2 Use the **configure crypto pki trustpoint** *ca-server-name* **authenticate** command to authenticate a trustpoint.

```
Device#configure crypto pki trustpoint ca-server-name authenticate
```

This command fetches the CA certificate from CA server automatically.

Step 3 Use the **configure crypto pki trustpoint** *ca-server-name* **key-size** *key-length* command to configure a private key size.

```
Device#configure crypto pki trustpoint ca-server-name key-size key-length
```

Step 4 Use the **configure crypto pki trustpoint** *ca-server-name* **subject-name** *name* `[Optional]` *2ltr-country-code state-name locality org-name org-unit email* command to configure the subject-name.

```
Device#configure crypto pki trustpoint ca-server-name subject-name name [Optional] 2ltr-country-code
 state-name locality org-name org-unit email
```

Step 5 Use the **configure crypto pki trustpoint** *ca-server-name* **enroll** command to enroll the trustpoint.

```
Device#configure crypto pki trustpoint ca-server-name enroll
```

Request the digitally signed certificate from the CA server.

Step 6 Use the **configure crypto pki trustpoint** *ca-server-name* **auto-enroll enable** *renew-percentage* command to enable auto-enroll.

```
Device#configure crypto pki trustpoint ca-server-name auto-enroll enable renew-percentage
```

Note
Use the **configure crypto pki trustpoint** *ca-server-name* **auto-enroll disable** command to disable the auto-enroll.

Step 7 [Optional] Use the **configure crypto pki trustpoint** *trustpoint-name* **delete** command to delete a trustpoint.

```
Device#configure crypto pki trustpoint trustpoint-name delete
```

Step 8 Use the **show crypto pki trustpoint** command to view the trustpoint summary.

```
Device#show crypto pki trustpoint
```

Step 9 Use the **show crypto pki trustpoint** *trustpoint-name* **certificate** command to view the details of the certificate for a specific trustpoint.

```
Device#show crypto pki trustpoint trustpoint-name certificate
```

Step 10 Use the **show crypto pki timers** command to view the public key infrastructure (PKI) timer information.

show crypto pki timers

```
Device#show crypto pki timers
```

Configure manual certificate enrollment using TFTP server

Procedure

Step 1 Specify the enrollment method.

Use the **configure crypto pki trustpoint** *ca-server-name* **enrollment tftp** *tftp-addr/file-name* command to retrieve the CA and client certificate for a trustpoint.

```
Device#configure crypto pki trustpoint ca-server-name enrollment tftp tftp-addr/file-name
```

Step 2 Use the **configure crypto pki trustpoint** *ca-server-name* **authenticate** command to authenticate a trustpoint manually.

```
Device#configure crypto pki trustpoint ca-server-name authenticate
```

This retrieves and authenticates the CA certificate from the specified TFTP server. If the file specification is included, the WGB adds the extension **.ca** to the specified filename.

Step 3 Use the **configure crypto pki trustpoint** *ca-server-name* **key-size** *key-length* command to configure a private key size.

```
Device#configure crypto pki trustpoint ca-server-name key-size key-length
```

Step 4 Use the **configure crypto pki trustpoint** *ca-server-name* **subject-name** *name* `[Optional]` *2ltr-country-code state-name locality org-name org-unit email* command to configure the subject-name.

```
Device#configure crypto pki trustpoint ca-server-name subject-name name [Optional] 2ltr-country-code
 state-name locality org-name org-unit email
```

Step 5 Use the **configure crypto pki trustpoint** *ca-server-name* **enroll** command to generate a private key and Certificate Signing Request (CSR).

```
Device#configure crypto pki trustpoint ca-server-name enroll
```

This generates certificate request and sends the request to the TFTP server. The filename to be written is appended with the **.req** extension.

Step 6 Use the **configure crypto pki trustpoint** *ca-server-name* **import certificate** command to import the signed certificate in WGB.

```
Device#configure crypto pki trustpoint ca-server-name import certificate
```

The console terminal uses TFTP to import a certificate and the WGB tries to get the approved certificate from the TFTP. The filename to be written is appended with the **.crt** extension.

Step 7 Use the **show crypto pki trustpoint** command to view the trustpoint summary.

```
Device#show crypto pki trustpoint
```

Step 8 Use the **show crypto pki trustpoint** *trustpoint-name* **certificate** command to view the content of the certificates that are created for a trustpoint.

```
Device#show crypto pki trustpoint trustpoint-name certificate
```

SSID configuration

SSID configuration consists of the following two parts:

1. Create an SSID profile, on page 11
2. Configuring Radio Interface for uWGB, on page 13

Create an SSID profile

Choose one of these authentication protocols to configure the SSID profile:

1. Configure an SSID profile using open authentication
2. Configure an SSID profile using PSK authentication
   - PSK WPA2 authentication
   - PSK Dot11r authentication, and
   - PSK Dot11w authentication.
3. Configure an SSID profile using Dot1x authentication

Configure an SSID profile using open authentication

Use the **configure ssid-profile** *ssid-profile-name* **ssid** *radio-serv-name* **authentication** *open* command to configure an SSID profile using open authentication.

```
Device#configure ssid-profile ssid-profile-name ssid radio-serv-name authentication open
```

Configure an SSID profile using PSK authentication

Choose one of these authentication protocols to configure an SSID profile using PSK authentication:

- configure an SSID profile using PSK WPA2 authentication
- configure an SSID profile using PSK Dot11r authentication, and
- configure an SSID profile using PSK Dot11w authentication .

Configure an SSID profile using PSK WPA2 authentication

Use the **configure ssid-profile** *ssid-profile-name* **ssid** *SSID_name* **authentication psk** *preshared-key* **key-management wpa2** command to configure an SSID profile using PSK WPA2 authentication.

```
Device#configure ssid-profile ssid-profile-name ssid SSID_name authentication psk
preshared-key key-management wpa2
```

Configure an SSID profile using PSK Dot11r authentication

Use the **configure ssid-profile** *ssid-profile-name* **ssid** *SSID_name* **authentication psk** *preshared-key* **key-management dot11r** command to configure an SSID profile using PSK Dot11r authentication.

```
Device#configure ssid-profile ssid-profile-name ssid SSID_name authentication psk
preshared-key key-management dot11r
```

Configure an SSID profile using PSK Dot11w authentication

Use the **configure ssid-profile** *ssid-profile-name* **ssid** *SSID_name* **authentication psk** *preshared-key* **key-management dot11w** command to configure an SSID profile using PSK Dot11w authentication

```
Device#configure ssid-profile ssid-profile-name ssid SSID_name authentication psk
preshared-key key-management dot11w
```

Configure an SSID profile using Dot1x authentication

Use the **configure ssid-profile** *ssid-profile-name* **ssid** *radio-serv-name* **authentication eap profile** *eap-profile-name* **key-management** {**dot11r** | **wpa2** | **dot11w** {**optional | required**} } command to configure an SSID profile using Dot1x authentication.

```
Device#configure ssid-profile ssid-profile-name ssid radio-serv-name authentication eap
profile eap-profile-name key-management { dot11r | wpa2 | dot11w { optional | required}}
```

Configure an SSID profile using Dot1x EAP-PEAP authentication

Here is an example that shows the configuration of an SSID profile using Dot1x EAP-PEAP authentication:

```
Device#configure dot1x credential c1 username wgbusr password cisco123456
Device#configure eap-profile p1 dot1x-credential c1
Device#configure eap-profile p1 method peap
Device#configure ssid-profile iot-peap ssid iot-peap authentication eap profile p1
key-management wpa2
```

Configuring Radio Interface for uWGB

- From the available two radio interfaces, before configuring WGB mode on one radio interface, configure the other radio interface to root-ap mode.

  Map a radio interface as root-ap by entering this command:

  # **configure dot11radio** *radio-interface* **mode root-ap**

  Example

  ```
  # configure dot11radio 0 mode root-ap
  ```

Note When an active SSID or EAP profile is modified, you need to reassociate the profile to the radio interface for the updated profile to be active.

- Map a radio interface to a WGB SSID profile by entering this command:

  # **configure dot11radio** *radio-interface* **mode uwgb** *uwgb-wired-client-mac-address* **ssid-profile** *ssid-profile-name*

- Configure a radio interface by entering this command:

  # **configure dot11radio** *radio-interface*{ **enable** | **disable** }

  Example

  ```
  # configure dot11radio 0 disable
  ```

Note After configuring the uplink to the SSID profile, we recommend you to disable and enable the radio for the changes to be active.

Note Only one radio or slot is allowed to operate in uWGB or WGB mode.

Configuring Workgroup Bridge Timeouts

The timer configuration CLIs are common for both WGB and uWGB. Use the following commands to configure timers:

- Configure the WGB association response timeout by entering this command:

  # **configure wgb association response timeout** *response-millisecs*

  The default value is 5000 milliseconds. The valid range is between 300 and 5000 milliseconds.

- Configure the WGB authentication response timeout by entering this command:

  # **configure wgb authentication response timeout** *response-millisecs*

  The default value is 5000 milliseconds. The valid range is between 300 and 5000 milliseconds.

- Configure the WGB EAP timeout by entering this command:

  # **configure wgb eap timeout** *timeout-secs*

  The default value is 3 seconds. The valid range is between 2 and 60 seconds.

- Configure the WGB bridge client response timeout by entering this command:

  # **configure wgb bridge client timeout** *timeout-secs*

  Default timeout value is 300 seconds. The valid range is between 10 and 1000000 seconds.

Flex Antenna Band Configuration

Flex antenna band configuration is supported on IW6300, ESW6300, and WP-WiFi6.

Use the following command to set antenna band to dual or single:

```
# configure wgb antenna band mode {dual|single}
```

Use the following command to check if WGB antenna band is set successfully:

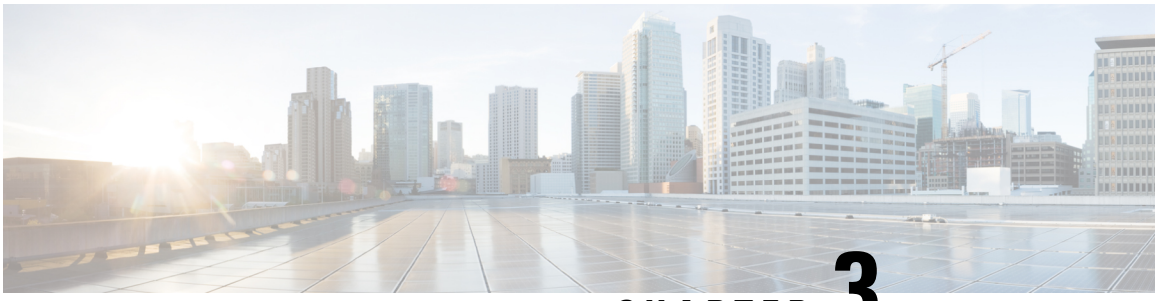
```
# show configuration | inc Band
```

For WP- WiFi6, use the following command to check WGB antenna band set by GPIO values. For single band: GPIO_34 : 0, GPIO_35 : 1. For dual band: GPIO_34 : 1, GPIO_35 : 0.

```
# show capwap client config | inc GPIO
GPIO_34                          : 1
GPIO_35                          : 0
```



Note IW6300 and ESW6300 do not support to check GPIO values.



CHAPTER 3

Configuring WGB

This chapter contains these topics:

- Configuring AP to WGB Mode, on page 15
- Configure IP address, on page 16
- Configure a Dot1X credential, on page 17
- Deauthenticate WGB wired client, on page 17
- Configure an EAP profile, on page 17
- Configure trustpoint manual enrollment for terminal, on page 18
- Configure trustpoint auto-enrollment for WGB , on page 19
- Configure manual certificate enrollment using TFTP server, on page 20
- SSID configuration, on page 21
- Configuring Workgroup Bridge Timeouts, on page 24
- Flex Antenna Band Configuration, on page 24

Configuring AP to WGB Mode

Cisco 802.11ac wave2 AP (IW6300 and ESW6300) and 802.11ax AP module (WP-WIFI6) are Cheetah OS (COS) based access points. The COS WGB function runs on the following image versions:

- **ap3g3-k9w8-tar.xxx.tar**
- **ap1g8-k9w8-tar.xxx.tar**

Make sure that you use the correct image version for WGB deployment.

- To configure a Cisco AP from Capwap mode to WGB mode, use the following command:

```
# ap-type workgroup-bridge

WGB is a wireless client that serve as nonroot ap for wired clients.
AP is the Master/CAPWAP AP, system will need a reboot when ap type is
changed to WGB. Do you want to proceed? (y/N):y
```

- To reverse the AP to Capwap mode, configure ap-type as Capwap by using the following command:

```
# ap-type capwap
```

Note Switching between EWC mode and WGB mode is not supported.

Configure IP address

Configure IPv4 address

- Use the **configure ap address** *ipv4 dhcp* command to configure IPv4 address using DHCP.

```
Device#configure ap address ipv4 dhcp
```

- Use the **configure ap address** *ipv4 static ipv4_addr netmask gateway* command to configure the static IPv4 address. By doing so, you can manage the device using a wired interface without an uplink connection.

```
Device#configure ap address ipv4 static ipv4_addr netmask gateway
```

Verify current IP configuration

Use **show ip interface brief** command to view the current IP address configuration.

```
Device#show ip interface brief
```

Configure IPv6 address

Use the **configure ap address ipv6 static** *ipv6_addr prefixlen [gateway]* command to configure the static IPv6 address. This configuration allows you to manage the AP through a wired interface without uplink connection.

```
Device#configure ap address ipv6 static ipv6_addr prefixlen [gateway]
```

Enable IPv6 auto configuration

Use the **configure ap address ipv6 auto-config enable** command to enable the IPv6 auto configuration on the AP.

```
Device#configure ap address ipv6 auto-config enable
```

Note

- Use the **configure ap address ipv6 auto-config disable** command to disable the IPv6 auto configuration on the AP.
- Use the **configure ap address ipv6 auto-config enable** command to enable IPv6 SLAAC. Note that SLAAC does not apply to CoS of WGB. This command configures IPv6 address with DHCPv6 instead of SLAAC.

Configure IPv6 address using DHCP

Use the **configure ap address** *ipv6 dhcp* command to configure IPv6 address using DHCP.

```
Device#configure ap address ipv6 dhcp
```

Verify current IP configuration

Use the **show ipv6 interface brief** command to verify current IP address configuration.

```
Device#show ipv6 interface brief
```

Configure a Dot1X credential

Use the **configure dot1x credential** *profile-name* **username** *name* **password** *pwd* command to configure Dot1x credential.

```
Device#configure dot1x credential profile-name username name password pwd
```

Verify WGB EAP Dot1x profile

Use the **show wgb eap dot1x credential profile** command to view the status of WGB EAP Dot1x profile.

```
Device#show wgb eap dot1x credential profile
```

Deauthenticate WGB wired client

Use the **clear wgb client** {**all** |**single** *mac-addr*} command to deauthenticate WGB wired client.

```
Device#clear wgb client {all |single mac-addr}
```

Configure an EAP profile

Perform these steps to configure an EAP profile:

1. Attach the Dot1x credential profile to the EAP profile.
2. Attach the EAP profile to the SSID profile.
3. Attach the SSID profile to the radio.

Procedure

Step 1 Use the **configure eap-profile** *profile-name* **method** {**fast** | **leap** | **peap** | **tls**} command to configure the EAP profile.

```
Device#configure eap-profile profile-name method { fast | leap | peap | tls}
```

Note
Choose an EAP profile method.

- fast
- peap, or
- tls.

Step 2 Use the **configure eap-profile** *profile-name* **trustpoint** {**default** | **name** *trustpoint-name*} command to attach the CA trustpoint for TLS. By default, the WGB uses the internal MIC certificate for authentication.

```
Device#configure eap-profile profile-name trustpoint { default | name trustpoint-name}
```

Step 3 Use the **configure eap-profile** *profile-name* **dot1x-credential** *profile-name* command to attach the dot1x-credential profile.

```
Device#configure eap-profile profile-name dot1x-credential profile-name
```

Step 4 [Optional] Use the **configure eap-profile** *profile-name* **delete** command to delete an EAP profile.

```
Device#configure eap-profile profile-name delete
```

Configure trustpoint manual enrollment for terminal

Procedure

Step 1 Use the **configure crypto pki trustpoint** *ca-server-name* **enrollment terminal** command to create a trustpoint in WGB.

```
Device#configure crypto pki trustpoint ca-server-name enrollment terminal
```

Step 2 Use the **configure crypto pki trustpoint** *ca-server-name* **authenticate** command to authenticate a trustpoint manually.

```
Device#configure crypto pki trustpoint ca-server-name authenticate
```

Enter the base 64 encoded CA certificate.

Enter **quit** to finish the certificate.

Note
If you use an intermediate certificate, import all the certificate chains in the trustpoint.

Example:

```
Device#configure crypto pki trustpoint demotp authenticate

Enter the base 64 encoded CA certificate.
....And end with the word "quit" on a line by itself....

-----BEGIN CERTIFICATE-----
[base64 encoded root CA certificate]
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
[base64 encoded intermediate CA certificate]
-----END CERTIFICATE-----
quit
```

Step 3 Use the **configure crypto pki trustpoint** *ca-server-name* **key-size** *key-length* command to configure a private key size.

```
Device#configure crypto pki trustpoint ca-server-name key-size key-length
```

Step 4 Use the **configure crypto pki trustpoint** *ca-server-name* **subject-name** *name* `[Optional]` *2ltr-country-code state-name locality org-name org-unit email* command to configure the subject-name.

```
Device#configure crypto pki trustpoint ca-server-name subject-name name [Optional] 2ltr-country-code
 state-name locality org-name org-unit email
```

Step 5 Use the **configure crypto pki trustpoint** *ca-server-name* **enroll** command to generate a private key and certificate signing request (CSR).

```
Device#configure crypto pki trustpoint ca-server-name enroll
```

Create the digitally signed certificate using the CSR output in the CA server.

Step 6 Use the **configure crypto pki trustpoint** *ca-server-name* **import certificate** command to import the signed certificate in WGB.

```
Device#configure crypto pki trustpoint ca-server-name import certificate
```

Enter the base 64 encoded CA certificate.

Enter **quit** to finish the certificate.

```
Device#quit
```

Step 7 [Optional] Use the **configure crypto pki trustpoint** *trustpoint-name* **delete** command to delete a trustpoint.

```
Device#configure crypto pki trustpoint trustpoint-name delete
```

Step 8 Use the **show crypto pki trustpoint** command to view the trustpoint summary.

```
Device#show crypto pki trustpoint
```

Step 9 Use the **show crypto pki trustpoint** *trustpoint-name* **certificate** command to view the content of the certificates that are created for a trustpoint.

```
Device#show crypto pki trustpoint trustpoint-name certificate
```

Configure trustpoint auto-enrollment for WGB

Procedure

Step 1 Use the **configure crypto pki trustpoint** *ca-server-name* **enrollment url** *ca-server-url* command to enroll a trustpoint in the WGB using the server URL.

```
Device#configure crypto pki trustpoint ca-server-name enrollment url ca-server-url
```

Step 2 Use the **configure crypto pki trustpoint** *ca-server-name* **authenticate** command to authenticate a trustpoint.

```
Device#configure crypto pki trustpoint ca-server-name authenticate
```

This command fetches the CA certificate from CA server automatically.

Step 3 Use the **configure crypto pki trustpoint** *ca-server-name* **key-size** *key-length* command to configure a private key size.

```
Device#configure crypto pki trustpoint ca-server-name key-size key-length
```

Step 4 Use the **configure crypto pki trustpoint** *ca-server-name* **subject-name** *name* `[Optional]` *2ltr-country-code state-name locality org-name org-unit email* command to configure the subject-name.

```
Device#configure crypto pki trustpoint ca-server-name subject-name name [Optional] 2ltr-country-code
 state-name locality org-name org-unit email
```

Step 5 Use the **configure crypto pki trustpoint** *ca-server-name* **enroll** command to enroll the trustpoint.

```
Device#configure crypto pki trustpoint ca-server-name enroll
```

Request the digitally signed certificate from the CA server.

Step 6 Use the **configure crypto pki trustpoint** *ca-server-name* **auto-enroll enable** *renew-percentage* command to enable auto-enroll.

```
Device#configure crypto pki trustpoint ca-server-name auto-enroll enable renew-percentage
```

Note
Use the **configure crypto pki trustpoint** *ca-server-name* **auto-enroll disable** command to disable the auto-enroll.

Step 7 [Optional] Use the **configure crypto pki trustpoint** *trustpoint-name* **delete** command to delete a trustpoint.

```
Device#configure crypto pki trustpoint trustpoint-name delete
```

Step 8 Use the **show crypto pki trustpoint** command to view the trustpoint summary.

```
Device#show crypto pki trustpoint
```

Step 9 Use the **show crypto pki trustpoint** *trustpoint-name* **certificate** command to view the details of the certificate for a specific trustpoint.

```
Device#show crypto pki trustpoint trustpoint-name certificate
```

Step 10 Use the **show crypto pki timers** command to view the public key infrastructure (PKI) timer information.

show crypto pki timers

```
Device#show crypto pki timers
```

Configure manual certificate enrollment using TFTP server

Procedure

Step 1 Specify the enrollment method.

Use the **configure crypto pki trustpoint** *ca-server-name* **enrollment tftp** *tftp-addr/file-name* command to retrieve the CA and client certificate for a trustpoint.

```
Device#configure crypto pki trustpoint ca-server-name enrollment tftp tftp-addr/file-name
```

Step 2 Use the **configure crypto pki trustpoint** *ca-server-name* **authenticate** command to authenticate a trustpoint manually.

```
Device#configure crypto pki trustpoint ca-server-name authenticate
```

This retrieves and authenticates the CA certificate from the specified TFTP server. If the file specification is included, the WGB adds the extension **.ca** to the specified filename.

Step 3 Use the **configure crypto pki trustpoint** *ca-server-name* **key-size** *key-length* command to configure a private key size.

```
Device#configure crypto pki trustpoint ca-server-name key-size key-length
```

Step 4 Use the **configure crypto pki trustpoint** *ca-server-name* **subject-name** *name* `[Optional]` *2ltr-country-code state-name locality org-name org-unit email* command to configure the subject-name.

```
Device#configure crypto pki trustpoint ca-server-name subject-name name [Optional] 2ltr-country-code
 state-name locality org-name org-unit email
```

Step 5 Use the **configure crypto pki trustpoint** *ca-server-name* **enroll** command to generate a private key and Certificate Signing Request (CSR).

```
Device#configure crypto pki trustpoint ca-server-name enroll
```

This generates certificate request and sends the request to the TFTP server. The filename to be written is appended with the **.req** extension.

Step 6 Use the **configure crypto pki trustpoint** *ca-server-name* **import certificate** command to import the signed certificate in WGB.

```
Device#configure crypto pki trustpoint ca-server-name import certificate
```

The console terminal uses TFTP to import a certificate and the WGB tries to get the approved certificate from the TFTP. The filename to be written is appended with the **.crt** extension.

Step 7 Use the **show crypto pki trustpoint** command to view the trustpoint summary.

```
Device#show crypto pki trustpoint
```

Step 8 Use the **show crypto pki trustpoint** *trustpoint-name* **certificate** command to view the content of the certificates that are created for a trustpoint.

```
Device#show crypto pki trustpoint trustpoint-name certificate
```

SSID configuration

SSID configuration consists of the following two parts:

1. Create an SSID profile, on page 11
2. Configuring Radio Interface for Workgroup Bridges, on page 23

Create an SSID profile

Choose one of these authentication protocols to configure the SSID profile:

1. Configure an SSID profile using open authentication
2. Configure an SSID profile using PSK authentication
   - PSK WPA2 authentication

- PSK Dot11r authentication, and
- PSK Dot11w authentication.

3. Configure an SSID profile using Dot1x authentication

Configure an SSID profile using open authentication

Use the **configure ssid-profile** *ssid-profile-name* **ssid** *radio-serv-name* **authentication** *open* command to configure an SSID profile using open authentication.

```
Device#configure ssid-profile ssid-profile-name ssid radio-serv-name authentication open
```

Configure an SSID profile using PSK authentication

Choose one of these authentication protocols to configure an SSID profile using PSK authentication:

- configure an SSID profile using PSK WPA2 authentication
- configure an SSID profile using PSK Dot11r authentication, and
- configure an SSID profile using PSK Dot11w authentication .

Configure an SSID profile using PSK WPA2 authentication

Use the **configure ssid-profile** *ssid-profile-name* **ssid** *SSID_name* **authentication psk** *preshared-key* **key-management wpa2** command to configure an SSID profile using PSK WPA2 authentication.

```
Device#configure ssid-profile ssid-profile-name ssid SSID_name authentication psk
preshared-key key-management wpa2
```

Configure an SSID profile using PSK Dot11r authentication

Use the **configure ssid-profile** *ssid-profile-name* **ssid** *SSID_name* **authentication psk** *preshared-key* **key-management dot11r** command to configure an SSID profile using PSK Dot11r authentication.

```
Device#configure ssid-profile ssid-profile-name ssid SSID_name authentication psk
preshared-key key-management dot11r
```

Configure an SSID profile using PSK Dot11w authentication

Use the **configure ssid-profile** *ssid-profile-name* **ssid** *SSID_name* **authentication psk** *preshared-key* **key-management dot11w** command to configure an SSID profile using PSK Dot11w authentication

```
Device#configure ssid-profile ssid-profile-name ssid SSID_name authentication psk
preshared-key key-management dot11w
```

Configure an SSID profile using Dot1x authentication

Use the **configure ssid-profile** *ssid-profile-name* **ssid** *radio-serv-name* **authentication eap profile** *eap-profile-name* **key-management** {**dot11r** | **wpa2** | **dot11w** {**optional | required**} } command to configure an SSID profile using Dot1x authentication.

```
Device#configure ssid-profile ssid-profile-name ssid radio-serv-name authentication eap
profile eap-profile-name key-management { dot11r | wpa2 | dot11w { optional | required}}
```

Configure an SSID profile using Dot1x EAP-PEAP authentication

Here is an example that shows the configuration of an SSID profile using Dot1x EAP-PEAP authentication:

```
Device#configure dot1x credential c1 username wgbusr password cisco123456
Device#configure eap-profile p1 dot1x-credential c1
Device#configure eap-profile p1 method peap
Device#configure ssid-profile iot-peap ssid iot-peap authentication eap profile p1
key-management wpa2
```

Configuring Radio Interface for Workgroup Bridges

- From the available two radio interfaces, before configuring WGB mode on one radio interface, configure the other radio interface to root-ap mode.

  Map a radio interface as root-ap by entering this command:

  # **configure dot11radio** *radio-interface* **mode root-ap**

  Example

  ```
  # configure dot11radio 0 mode root-ap
  ```

Note When an active SSID or EAP profile is modified, you need to reassociate the profile to the radio interface for the updated profile to be active.

- Map a radio interface to a WGB SSID profile by entering this command:

  # **configure dot11radio** *radio-interface* **mode wgb ssid-profile** *ssid-profile-name*

  Example

  ```
  # configure dot11radio 1 mode wgb ssid-profile psk_ssid
  ```

- Configure a radio interface by entering this command:

  # **configure dot11radio** *radio-interface*{ **enable** | **disable** }

  Example

  ```
  # configure dot11radio 0 disable
  ```

Note After configuring the uplink to the SSID profile, we recommend you to disable and enable the radio for the changes to be active.

Note Only one radio or slot is allowed to operate in WGB mode.

Configuring Workgroup Bridge Timeouts

The timer configuration CLIs are common for both WGB and uWGB. Use the following commands to configure timers:

- Configure the WGB association response timeout by entering this command:

  # **configure wgb association response timeout** *response-millisecs*

  The default value is 5000 milliseconds. The valid range is between 300 and 5000 milliseconds.

- Configure the WGB authentication response timeout by entering this command:

  # **configure wgb authentication response timeout** *response-millisecs*

  The default value is 5000 milliseconds. The valid range is between 300 and 5000 milliseconds.

- Configure the WGB EAP timeout by entering this command:

  # **configure wgb eap timeout** *timeout-secs*

  The default value is 3 seconds. The valid range is between 2 and 60 seconds.

- Configure the WGB bridge client response timeout by entering this command:

  # **configure wgb bridge client timeout** *timeout-secs*

  Default timeout value is 300 seconds. The valid range is between 10 and 1000000 seconds.

Flex Antenna Band Configuration

Flex antenna band configuration is supported on IW6300, ESW6300, and WP-WiFi6.

Use the following command to set antenna band to dual or single:

```
# configure wgb antenna band mode {dual|single}
```

Use the following command to check if WGB antenna band is set successfully:

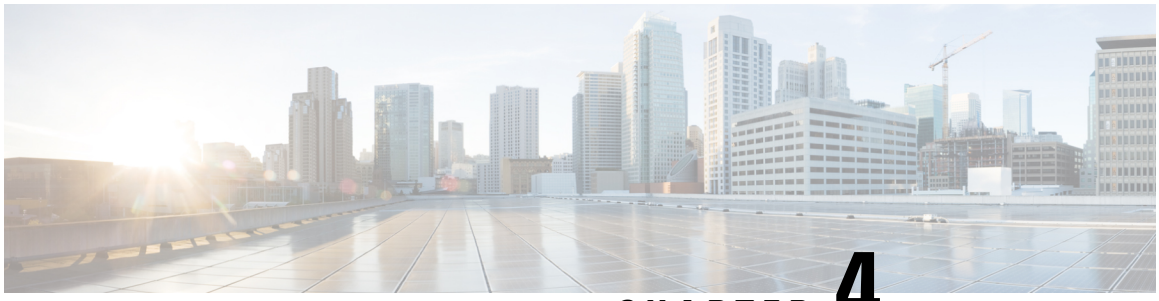
```
# show configuration | inc Band
```

For WP- WiFi6, use the following command to check WGB antenna band set by GPIO values. For single band: GPIO_34 : 0, GPIO_35 : 1. For dual band: GPIO_34 : 1, GPIO_35 : 0.

```
# show capwap client config | inc GPIO
GPIO_34                          : 1
GPIO_35                          : 0
```



Note IW6300 and ESW6300 do not support to check GPIO values.



CHAPTER 4

Workgroup Bridge FAQ

This chapter provides information on the most common questions asked about Cisco Workgroup Bridges.

- Workgroup Bridge FAQ, on page 25

Workgroup Bridge FAQ

- **Q. What is a Workgroup Bridge?**

  A. A workgroup bridge (WGB) is a special mode on a Cisco access point that can associate to wireless access point as a client and provide wireless connectivity for wired devices that connect to its Ethernet port.

- **Q. Can Workgroup Bridge associate with non-Cisco access point?**

  A. WGB sends Internet Access Point Protocol (IAPP) messages to the wireless access point to inform it about the MAC addresses of wired clients relayed through the workgroup bridge radio. When the access point is not a Cisco access point, these messages are not understood, so standard Workgroup Bridge cannot associate to non-Cisco access point, a special role of WGB – universal WGB was introduced to allow it to associate with non-Cisco access point.

- **Q. What is an universal Workgroup Bridge?**

  A. The universal WGB is able to interoperate with non-Cisco access points using uplink radio MAC address, thus the universal workgroup bridge role supports only one wired client. When works as universal WGB, the universal WGB is transparent and is not managed.

- **Q. What are the typical applications for a Workgroup Bridge?**

  A.

  - Stretching wireless infrastructure to wired-only clients
  - Deployments where it is not feasible or practical to run a cable to the wired device
  - In-vehicle deployments, where the WGB provides connectivity from autonomous guided vehicles, mining trucks etc. to a wireless network

- **Q. Which Cisco access point supports WGB today?**

  A. Cisco IW3702, Embedded AP803 module in IR829 are IOS based access points and support IOS WGB. The IOS WGB function can be running with an autonomous image, such as

ap3g2-k9w7-tar.xxx.tar. Cisco 802.11ac wave2 APs (ESW6300, IW6300, 1560, 2800, 3800) and 802.11ax AP module (WP-WIFI6) are Cheetah OS (COS) based access points. The COS WGB function runs on image version **ap3g3-k9w8-tar.xxx.tar**, **ap1g7-k9w8-tar.xxx.tar**, or **ap1g8-k9w8-tar.xxx.tar**.

- **Q. How many clients can be supported by a WGB?**

  A. A maximum of 20 wired clients are supported behind a Cisco WGB device, which is the max number allowed on wireless LAN controller (WLC).

- **Q. Does a WGB support multiple VLANs in it?**

  A. Yes, it is possible to associate WGB (WGB BVI interface) as a Native VLAN and have wired clients configured behind a dot1q switch associated to different (non-Native) VLANs.

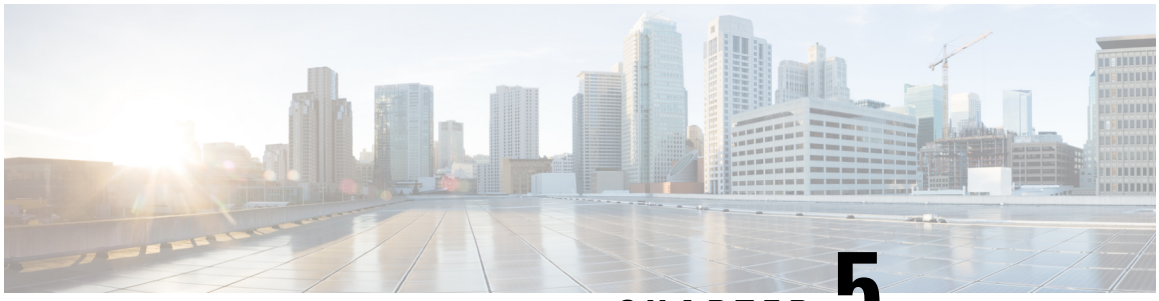
- **Q. How is WGB mode different from bridge (mesh) mode?**

  A. Bridge (mesh) mode is only suitable for stationary use case, while WGB as a wireless client can be deployed for both stationary and on the move use case.

- **Q. What are the key questions to ask before a Workgroup Bridge deployment?**

  A. A. In general, you should know the details of the application and WiFi infrastructure.

  - What is the application that will run on top of the wireless infrastructure? Is it latency and jitter sensitive application?
  - What is the needed bandwidth for the application?
  - What is the roaming delay tolerance?
  - Can the application handle properly network disconnections? Is there an additional backup mechanism?
  - Can the application handle packet loss properly? (Even on the best wireless design, you must expect a percentage of packet loss.)
  - Has site survey been conducted properly? Is the RF coverage good enough to support the required application bandwidth?



CHAPTER 5

Troubleshooting

This chapter provides information about troubleshooting.

- Debug Commands, on page 27
- WGB Show Commands, on page 27
- uWGB Show Commands, on page 28
- WGB Debug Examples, on page 29

Debug Commands

- **debug wgb uplink state-machine {events | info | error | critical | all}**
- **debug wgb uplink scan {events | info | error | critical | all}**
- **debug wgb uplink security {events | info | error | critical | all}**
- **debug wgb uplink configuration {events | info | error | critical | all}**

WGB Show Commands

Use these commands to check WGB configurations:

- **show running-config**
- **show wgb dot11 association**

```
wgb#show wgb dot11 associations
Uplink Radio ID             : 1
Uplink Radio MAC            : E8:EB:34:AE:AA:EF
SSID Name                   : B-thes-PSK
Parent AP Name              : ap9120-c01
Parent AP MAC               : 2C:57:41:93:0B:2C
Uplink State                : CONNECTED
Auth Type                   : PSK
Key management Type         : WPA2
Dot11 type                  : 11ax
Channel                     : 36
Bandwidth                   : 20 MHz
Current Datarate            : 6 Mbps
Max Datarate                : 286 Mbps
RSSI                        : 21
```

```
IP                          : 192.168.23.195/24
Default Gateway             : 192.168.23.1
DNS Server1                 : 192.168.71.2
Domain                      : iottest.local
IPV6                        : 2001:dead:beef:2103:de33:3013:8126:a39b/128
Assoc timeout               : 5000 Msec
Auth timeout                : 5000 Msec
Dhcp timeout                : 60 Sec
```

- **show wgb ssid**

```
wgb# show wgb ssid

Configured SSIDs details:
SSID-Profile                    SSID                              Authentication
================================================================================
sp                              sp                                PSK
B-thes-OP                       B-thes-OP                         OPEN
B-thes-1X                       B-thes-1X                         DOT1X
B-thes-PSK                      B-thes-PSK                        PSK

Connected SSIDs details:
  Radio ID       : 1
  Radio Mode     : WGB
  BSSID          : 2C:57:41:93:0B:2C
  SSID           : B-thes-PSK
  Authentication : PSK
```

- **show wgb bridge**

```
wgb# show wgb bridge
              mac vap   port vlan_id       seen_ip confirm_ago fast_brg
6C:2B:FC:2C:18:37   0 wired0       0 100.100.220.31   11.320000     true
00:2B:2C:07:3F:11   0 wired0       0       0.0.0.0    1.960000     true
```

- **show wgb packet statistics**

uWGB Show Commands

Use these commands to check uWGB configurations:

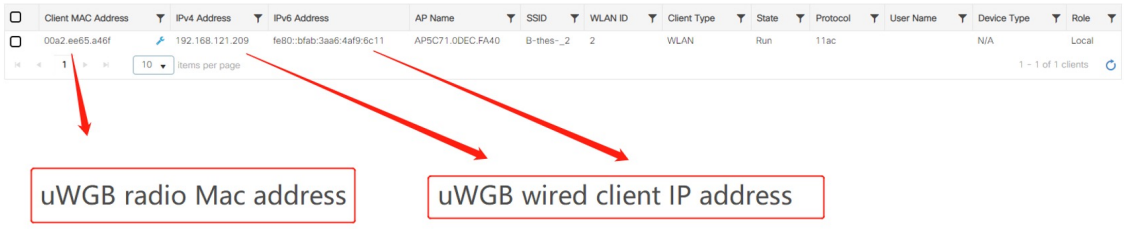
- **show running-config**
- **show wgb dot11 association**
- **show wgb ssid**
- **show wgb packet statistics**

The following example shows the **show wgb dot11 association** command output. Note that the current state should be "uwgb" if uclient is active.

```
wgb#show wgb dot11 associations
Uplink Radio ID             : 1
Uplink Radio MAC            : 00:A2:EE:65:A4:6F
SSID Name                   : B-thes-_3
Parent AP MAC               : 3C:41:0E:3B:02:0D
Uplink State                : CONNECTED
Auth Type                   : DOT1X
EAP Method Name             : FAST
Key management Type         : WPA2
```

```
Uclient mac                 : 00:50:56:A1:01:DC
Current state               : UWGB
Uclient timeout             : 60 Sec
Dot11 type                  : 11ac
Channel                     : 36
RSSI                        : 46
IP                          : 0.0.0.0
IPV6                        : ::/128
Assoc timeout               : 5000 Msec
Auth timeout                : 5000 Msec
Dhcp timeout                : 60 Sec
```

uWGB checkpoints on AP



WGB Debug Examples

This section provides WGB debug examples.

Figure 2: Preconfiguration for Hostapd and DOT11_UPLINK_CONFIG Before WGB Uplink Mode Started

```
DOT11_UPLINK_EV: Uplink state changed [DOT11_UPLINK_STOPPED] to [DOT11_UPLINK_SCAN_START]
DOT11_UPLINK_EV: Uplink state changed [DOT11_UPLINK_SCAN_START] to [DOT11_UPLINK_STOPPED]
DOT11_UPLINK_EV: DHCP: client process stopped
DOT11_UPLINK_CONFIG: Added WLAN Index : 0 to db
DOT11_UPLINK_CONFIG: Added Mac addr: E8:EB:34:AE:AA:EF to db
DOT11_UPLINK_EV: DHCP: clien--More--
WGB_UPLINK_SEC: WPA supplicant configuration file /etc/wpa_supplicant.conf created
systemd[1]: Starting WPA supplicant process...
systemd[1]: Started WPA supplicant process.
DOT11_UPLINK_CONFIG: WPAS started
WGB_UPLINK_SEC: wpas socket exists, but does not allow connections - assuming it was left over from process termination
WGB_UPLINK_SEC: Successfully replaced leftover socket '/var/run/wpa_supplicant/local'
WGB_UPLINK_SEC: wcp/wgb_sec :: Dot11UplinkSecurity: Error connecting to wpa supplicant socket for remote path /var/run/wpa_supplicant/wbridge1
WGB_UPLINK_SEC:
DOT11_UPLINK_CONFIG: Radio[1] configured for standard: 7
licant: FIPS Mode = disabled **
hostapd:wpa_supplicant v2.9
hostap version
hostapd:random: Trying to read entropy from /dev/random
hostapd:Successfully initialized wpa_supplicant
hostapd:Initializing interface 'wbridge1' conf '/etc/wpa_supplicant.conf' driver 'wpas' ctrl_interface '/var/run/wpa_supplicant' bridge 'N/A'
hostapd:Configuration file '/etc/wpa_supplicant.conf' -> '/etc/wpa_supplicant.conf'
hostapd:Reading configuration file '/etc/wpa_supplicant.conf'
hostapd:ctrl_interface='/var/run/wpa_supplicant'
hostapd:update_config=1
hostapd:eapol_version=1
hostapd:ap_scan=0
hostapd:fast_reauth=1
hostapd:Line: 6 - start of a new network block
ssid - hexdump_ascii(len=10):
     42 2d 74 68 65 73 2d 50 53 4b                     B-thes-PSK
hostapd:proto: 0x2
hostapd:key_mgmt: 0x2
PSK (ASCII passphrase) - hexdump_ascii(len=8): [REMOVED]
hostapd:pairwise: 0x10
hostapd:group: 0x10
PSK (from passphrase) - hexdump(len=32): [REMOVED]
hostapd:Priority group 0
hostapd:   id=0 ssid='B-thes-PSK'
hostapd:driver_wpas: wpas_init
DOT11_UPLINK_CONFIG: current power level: 1
DOT11_UPLINK_CONFIG: set tx power level: 1
DOT11_UPLINK_CONFIG: 3 Antennas configured
DOT11_UPLINK_CONFIG: wcp/wgb_config :: Dot11UplinkConfig: push antenna config to driver TxAntenna 7 RxAntenna 7
```

Figure 3: WGB Uplink Mode Started -> Scan Started

```
**** Uplink is disabled  ****
DOT11_UPLINK_DRIVER[1]: WGB uplink mode started
DOT11_UPLINK_EV: Scan Started
DOT11_UPLINK_EV: Uplink state changed [DOT11_UPLINK_STOPPED] to [DOT11_UPLINK_SCAN_START]
DOT11_UPLINK_SCAN: Uplink Scan Started in Dot11Radio1.
```

Figure 4: Parent selected

```
DOT11_UPLINK_EV: Calling RSSI get for 00:00:00:00:00:00
DOT11_UPLINK_EV: Last pkt RSSI: 0 0 0 0
DOT11_UPLINK_EV: Avg  RSSI: 0 0 0 0
DOT11_UPLINK_EV: parent_rssi: 0, configured low rssi: -70
DOT11_UPLINK_EV: Found BSSID 2c:57:41:93:0B:2C for SSID B-thes-PSK on channel 52
DOT11_UPLINK_EV: RSSI: -30
DOT11_UPLINK_EV: SNR: 35
DOT11_UPLINK_EV: load: 0
DOT11_UPLINK_EV: hops: 0
DOT11_UPLINK_EV: distance: 0
DOT11_UPLINK_EV: num assoc: 0
DOT11_UPLINK_EV: radio_type: 7
DOT11_UPLINK_EV: mdId: 0
DOT11_UPLINK_EV: ft_cap_policy: 0
DOT11_UPLINK_EV: channel width: 40
DOT11_UPLINK_EV: update_scan_result: it != NULL
DOT11_UPLINK_EV: update_scan_result: expired_time not initialized
DOT11_UPLINK_EV: update_scan_result:  return
DOT11_UPLINK_EV: Scan Finished. Total BSS found 1.
DOT11_UPLINK_EV: find best ap bssid
DOT11_UPLINK_EV: Current BSS--More--
DOT11_UPLINK_EV: same node. skipping...
DOT11_UPLINK_EV: Best BSSID: 2C:57:41:93:0B:2C
DOT11_UPLINK_EV: tgr_method: over-the-air
DOT11_UPLINK_SCAN: Uplink Scan stopped in Dot11Radio1
DOT11_UPLINK_EV: Uplink state changed [DOT11_UPLINK_SCAN_START] to [DOT11_UPLINK_SCAN_DONE]
```

Figure 5: Authenticating

```
DOT11_UPLINK_EV: existing channel 0, target channel 52
DOT11_UPLINK_EV: Starting Connection (uplink)addr1[E8:EB:34:AE:AA:EF], (bssid)addr2[2C:57:41:93:0B:2C]
WGB_UPLINK_SEC: WPA supplicant configuration file /etc/wpa_supplicant.conf created
WGB_UPLINK_SEC: wpas socket exists, but does not allow connections - assuming it was left over from process termination
WGB_UPLINK_SEC: Successfully replaced leftover socket '/var/run/wpa_supplicant/local'
WGB_UPLINK_SEC: start socket to WPA supplicant
WGB_UPLINK_SEC: Re configuration command to wpa_supplicant sent successfully..
WGB_UPLINK_SEC: Bad response for RECONFIGURE from wpa_supplicant
WGB_UPLINK_SEC: Sending init--More--
WGB_UPLINK_SEC: Bad response for EAPOLINIT from wpa_supplicant
WGB_UPLINK_SEC: New roamed parent : 2C:57:41:93:0B:2C
DOT11_UPLINK_EV: Uplink state changed [DOT11_UPLINK_SCAN_DONE] to [DOT11_UPLINK_AUTHENTICATING]
WGB Classifier: Dot11UplinkClassifier: Downstream packet fc b0 Len 30 MAC 2C:57:41:93:0B:2C
WGB Classifier: Dot11UplinkClassifier: Tx sent to Uplink Access-Point b0
DOT11_UPLINK_EV: Auth request sent!
WGB_UPLINK_SEC: Uplink client state is invalid(3).Dropping the EAP packet.
hostapd:wpa_ether_send() res=1
hostapd:wbridge1: Control interface command 'RECONFIGURE'
hostapd:Reading configuration file '/etc/wpa_supplicant.conf'
hostapd:ctrl_interface='/var/run/wpa_supplicant'
hostapd:update_config=1
hostapd:eapol_version=1
hostapd:ap_scan=0
hostapd:fast_reauth=1
hostapd:Line: 6 - start of a new network block
ssid - hexdump_ascii(len=10):
     42 2d 74 68 65 73 2d 50 53 4b                     B-thes-PSK
BSSID - hexdump(len=6): 2c 57 41 93 0b 2c
hostapd:proto: 0x2
hostapd:key_mgmt: 0x2
PSK (ASCII passphrase) - hexdump_ascii(len=8): [REMOVED]
hostapd:pairwise: 0x10
hostapd:group: 0x10
WGB Classifier: Dot11UplinkClassifier: Rx sent to WGB Uplink b0
DOT11_UPLINK_EV: Auth Response (uplink)addr1[E8:EB:34:AE:AA:EF], (bssid)addr2[2C:57:41:93:0B:2C] status code [0]
```

Figure 6: Authenticating -> Associating

```
DOT11_UPLINK_EV: Uplink state changed [DOT11_UPLINK_AUTHENTICATING] to [DOT11_UPLINK_ASSOCIATING]
DOT11_UPLINK_EV: Assoc. Req. addr1[E8:EB:34:AE:AA:EF], (bssid)addr2[2C:57:41:93:0B:2C]
DOT11_UPLINK_EV: set_ht_cap_ie_fields: Adding HT Information element to Assoc request slot_id 1
DOT11_UPLINK_EV: set_vht_cap_ie_fields: Adding VHT Information element to Assoc request slot_id 1
DOT11_UPLINK_EV: set_he_cap_ie_fields: Adding HE Information element to Assoc request slot_id 1
DOT11_UPLINK_EV: Added system name : wgb in assoc req.
WGB Classifier: Dot11UplinkClassifier: Downstream packet fc 0 Len 219 MAC 2C:57:41:93:0B:2C
WGB Classifier: Dot11UplinkClassifier: Tx sent to Uplink Access-Point 0
DOT11_UPLINK_EV: Sent Assoc. Req. (uplink)addr1[E8:EB:34:AE:AA:EF], (bssid)addr2[2C:57:41:93:0B:2C]
WGB Classifier: Dot11UplinkClassifier: Rx sent to WGB Uplink 10
DOT11_UPLINK_EV: Assoc Response (uplink)addr1[E8:EB:34:AE:AA:EF], (bssid)addr2[2C:57:41:93:0B:2C] status code [0]
```

Figure 7: Associating -> EAPoL Key

```
Apr  9 05:14:44 kernel: [+04/09/2021 05:14:44.5410] DOT11_UPLINK_EV: Eap_timer scheduled in 3000 msec
Apr  9 05:14:44 kernel: [+04/09/2021 05:14:44.5410] WGB Classifier: Dot11UplinkClassifier: Upstream packet from if 0 Len 135 Dest MAC E8:EB:34:AE:AA:EF AP MAC 2C:57:41:93:0B:2C ethertype 8e88
Apr  9 05:14:44 kernel: [+04/09/2021 05:14:44.5410] WGB_UPLINK_SEC: EAPOL packet from parent =2C:57:41:93:0B:2C len = 135
Apr  9 05:14:44 kernel: [+04/09/2021 05:14:44.5420] WGB_UPLINK_SEC: Sent EAPOL packet EAPOLPKT to wpa_supplicant response=OK
Apr  9 05:14:44 kernel: [+04/09/2021 05:14:44.5420] hostapd:WPAS State : 6
Apr  9 05:14:44 kernel: [+04/09/2021 05:14:44.5420] hostapd:wbridge1: Event EAPOL_RX (23) received
Apr  9 05:14:44 kernel: [+04/09/2021 05:14:44.5420] hostapd:wbridge1: RX EAPOL from 2c:57:41:93:0b:2c
Apr  9 05:14:44 kernel: [+04/09/2021 05:14:44.5420] RX EAPOL - hexdump(len=121): 02 03 00 75 02 00 8a 00 10 00 00 00 00 00 00 e3 b3 3a 38 2b ce 1e 4c b4 63 e6 e3 0c 2d 5a bd 50 b4 69 93 c9 8f 5c eb e3 4b b9 f6 3f 29 4a f8 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16 dd 14 00 0f ac 04 e5 8c 29 ef 5e 74 f0 db 83 32 50 8d 96 74 1a b3
Apr  9 05:14:44 kernel: [+04/09/2021 05:14:44.5420] hostapd:wbridge1: Setting authentication timeout: 10 sec 0 usec
Apr  9 05:14:44 kernel: [+04/09/2021 05:14:44.5420] hostapd:wbridge1: IEEE 802.1X RX: version=2 type=3 length=117
len=121): 02 03 00 75 02 00 8a 00 10 00 00 00 00 00 00 e3 b3 3a 38 2b ce 1e 4c b4 63 e6 e3 0c 2d 5a bd 50 b4 69 93 c9 8f 5c eb e3 4b b9 f6 3f 29 4a f8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16 dd 14 00 0f ac 04 e5 8c 29 ef 5e 74 f0 db 83 32 50 8d 96 74 1a b3
Apr  9 05:14:44 kernel: [+04/09/2021 05:14:44.5420] hostapd:wbridge1:   EAPOL-Key type=2
Apr  9 05:14:44 kernel: [+04/09/2021 05:14:44.5420] hostapd:wbridge1:   key_info 0x8a (ver=2 keyidx=0 rsvd=0 Pairwise Ack)
Apr  9 05:14:44 kernel: [+04/09/2021 05:14:44.5420] hostapd:wbridge1:   key_length=16 key_data_length=22
Apr  9 05:14:44 kernel: [+04/09/2021 05:14:44.5420]   replay_counter - hexdump(len=8): 00 00 00 00 00 00 00 09
Apr  9 05:14:44 kernel: [+04/09/2021 05:14:44.5420]   key_nonce - hexdump(len=32): a3 b3 3a 38 2b ce 1e 4c b4 63 e6 e3 0c 2d 5a bd 50 b4 69 93 c9 8f 5c eb e3 4b b9 f6 3f 29 4a f8
Apr  9 05:14:44 kernel: [+04/09/2021 05:14:44.5420]   key_iv - hexdump(len=16): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr  9 05:14:44 kernel: [+04/09/2021 05:14:44.5420]   key_rsc - hexdump(len=8): 00 00 00 00 00 00 00 00
Apr  9 05:14:44 kernel: [+04/09/2021 05:14:44.5420]   key_id (reserved) - hexdump(len=8): 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0020]   key_mic - hexdump(len=16):--More--
```

Figure 8: 4-way handshake

```
hostapd:wbridge1: State: ASSOCIATED -> 4WAY_HANDSHAKE
hostapd:wbridge1: WPA: RX message 1 of 4-Way Handshake from 2c:57:41:93:0b:2c (ver=2)
RSN: msg 1/4 key data - hexdump(len=22): dd 14 00 0f ac 04 e5 8c 29 ef 5e 74 f0 db 83 32 50 8d 96 74 1a b3
WPA: PMKID in EAPOL-Key - hexdump(len=22): dd 14 00 0f ac 04 e5 8c 29 ef 5e 74 f0 db 83 32 50 8d 96 74 1a b3
RSN: PMKID from Authenticator - hexdump(len=16): e5 8c 29 ef 5e 74 f0 db 83 32 50 8d 96 74 1a b3
hostapd:wbridge1: RSN: no matching PMKID found
PMKId in M1 :   - hexdump(len=16): e5 8c 29 ef 5e 74 f0 db 83 32 50 8d 96 74 1a b3
hostapd:Get randomness: len=32 entropy=2
WPA: Renewed SNonce - hexdump(len=32): b1 21 6d 62 dc 8c f7 1a 8d d2 77 09 1b 21 81 05 9d 7a 59 5d e8 c8 f4 6a dc 88 cd 23 a9 10 fb 83
hostapd:WPA: PTK derivation using PRF(SHA1)
hostapd:WPA: PTK derivation --More--
WPA: Nonce1 - hexdump(len=32): b1 21 6d 62 dc 8c f7 1a 8d d2 77 09 1b 21 81 05 9d 7a 59 5d e8 c8 f4 6a dc 88 cd 23 a9 10 fb 83
WPA: Nonce2 - hexdump(len=32): a3 b3 3a 38 2b ce 1e 4c b4 63 e6 e3 0c 2d 5a bd 50 b4 69 93 c9 8f 5c eb e3 4b b9 f6 3f 29 4a f8
WPA: PMK - hexdump(len=32): [REMOVED]
WPA: PTK - hexdump(len=48): [REMOVED]
WPA: KCK - hexdump(len=16): [REMOVED]
WPA: KEK - hexdump(len=16): [REMOVED]
WPA: TK - hexdump(len=16): [REMOVED]
```

```
WPA: WPA IE for msg 2/4 - hexdump(len=22): 30 14 01 00 00 0f ac 04 01 00 00 0f ac 04 01 00 00 0f ac 02 3c 00
WPA: Replay Counter - hexdump(len=8): 00 00 00 00 00 00 00 00
hostapd:wbridge1: WPA: Sending EAPOL-Key 2/4
hostapd:WPA: Send EAPOL-Key frame to 2c:57:41:93:0b:2c ver=2 mic_len=16 key_mgmt=0x2
hostapd:WPA: EAPOL-Key MIC using HMAC-SHA1
WPA: KCK - hexdump(len=16): [REMOVED]
```

```
hostapd:wbridge1: State: 4WAY_HANDSHAKE -> 4WAY_HANDSHAKE
hostapd:wbridge1: WPA: RX message 3 of 4-Way Handshake from 2c:57:41:93:0b:2c (ver=2)
WPA: IE KeyData - hexdump(len=48): 30 14 01 00 00 0f ac 04 01 00 00 0f ac 04 01 00 00 0f ac 02 28 00 dd 16 00 0f ac 01 01 00 ef 78 ea bf cf a7 eb a3 c6 ea cb 52 52 6d eb ef dd 00
WPA: RSN IE in EAPOL-Key - hexdump(len=22): 30 14 01 00 00 0f ac 04 01 00 00 0f ac 04 01 00 00 0f ac 02 28 00
```

```
hostapd:wbridge1: WPA: Sending EAPOL-Key 4/4
hostapd:WPA: Send EAPOL-Key frame to 2c:57:41:93:0b:2c ver=2 mic_len=16 key_mgmt=0x2
hostapd:WPA: EAPOL-Key MIC using HMAC-SHA1
WPA: KCK - hexdump(len=16): [REMOVED]
WPA: Derived Key MIC - hexdump(len=16): 27 fb e1 4c 9d c4 d4 23 a2 7e 6b bb 11 8e f7 79
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
WGB_UPLINK_SEC: Received EAPOL from wpas source : E8:EB:34:AE:AA:EF, dest : 2C:57:41:93:0B:2C
WGB_UPLINK_SEC: Total len of packet = 113 Dump = 2c574193 0b2ce8eb 34aeaaef 888e0103 005f0203 0a00
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000027
fbe14c9d c4d423a2 7e6bbb11 8ef77900 00
hostapd:wbridge1: WPA: Installing PTK to the driver
```

Figure 9: GTK derived

```
hostapd:wbridge1: State: 4WAY_HANDSHAKE -> GROUP_HANDSHAKE
RSN: received GTK in pairwise handshake - hexdump(len=18): [REMOVED]
WPA: Group Key - hexdump(len=16): [REMOVED]
hostapd:wbridge1: WPA: Installing GTK to the driver (keyidx=1 tx=0 len=16)
WPA: RSC - hexdump(len=6): 00 00 00 00 00 00
hostapd: wpas_set_key change ik_macaddr to bssid
```

Figure 10: DOT11 UPLINK ESTABLISHED

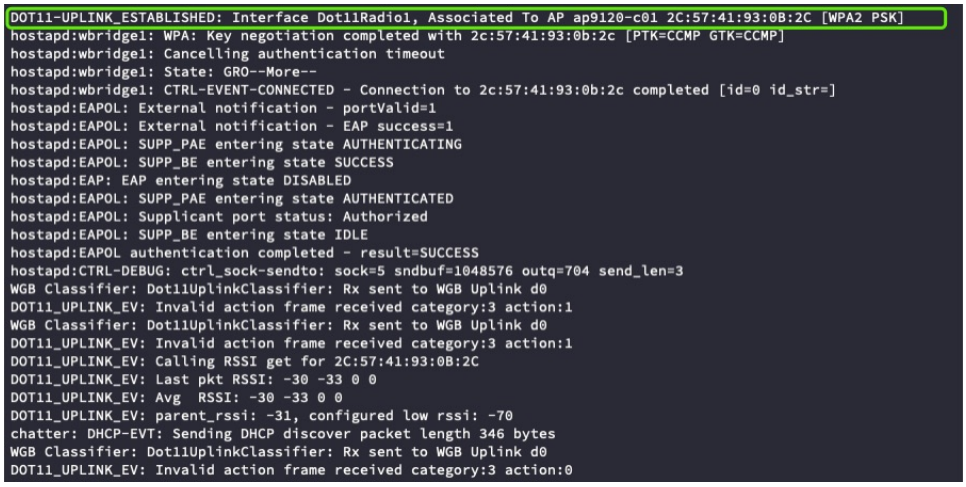


Figure 11: DHCP -> Connected State

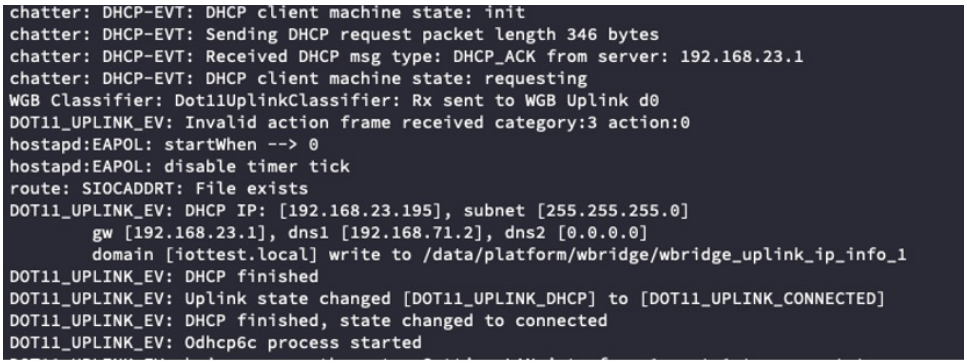


Figure 12: DHCPv6

