Understand the AP Join Process with the Catalyst 9800 WLC

Contents

Introduction

Prerequisites

- Requirements
- Components Used

Background Information

- CAPWAP Session Establishment
- DTLS Session Establishment
- Wireless LAN Controller Discovery Methods
- Wireless LAN Controller Election
- CAPWAP State Machine
  - CAPWAP State: Discovery
  - CAPWAP State: DTLS Setup.
  - CAPWAP State: Join
  - CAPWAP State: Image Data
  - CAPWAP State: Configure
  - CAPWAP State: Run

Configure

- Static WLC Election
- Enabling Telnet/SSH Access to the AP
- Data Link Encryption

Verify

Troubleshoot

- Known Issues
- WLC GUI Checks
- Commands
  - From the WLC
  - From Wave 2 and Catalyst 11ax APs
  - From Wave 1 APs
- Radioactive Traces

Introduction

This document describes in detail the AP Join Process with the Cisco Catalyst 9800 WLC.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Basic understanding of the Control and Provisioning Wireless Access Points (CAPWAP)
- Basic understanding of the usage of a Wireless Lan Controller (WLC)

Components Used

The information in this document is based on these software and hardware versions:

- Catalyst 9800-L WLC, Cisco IOS® XE Cupertino 17.9.3
- Catalyst 9120AXE Access Point

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

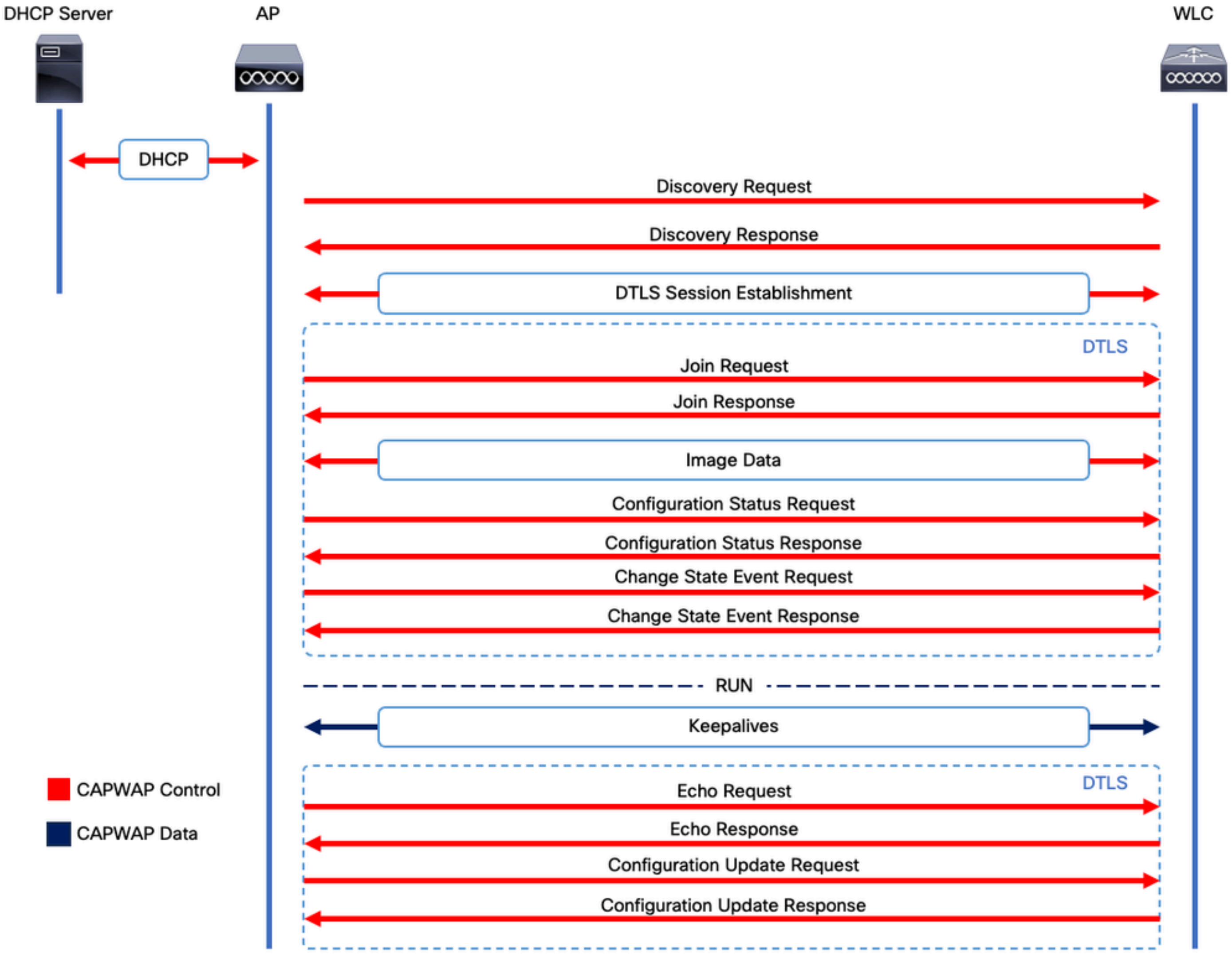
CAPWAP Session Establishment

Control And Provisioning Wireless Access Point (CAPWAP) is the protocol that provides the transport mechanism used by Access Points (APs) and Wireless LAN Controllers (WLCs) to exchange control and data plane information over a secure communication tunnel (for CAPWAP Control).

In order to elaborate on the AP Join process, it is important that you understand the Control And Provisioning Wireless Access Point (CAPWAP) session establishment process.

Please keep in mind that the AP needs to have an IP address before being able to start the CAPWAP Process. If the AP does not have an IP address, it does not initiate the **CAPWAP Session Establishment Process.**

1. Access Point sends a **Discovery Request**. See the WLC Discovery Methods section for more information on this
2. WLC sends a **Discovery Response**
3. DTLS session establishment. After this, all messages after this are encrypted and are shown as DTLS application data packets in any packet analysis tool.
4. Access Point sends a **Join Request**
5. WLC sends a **Join Response**
6. AP performs an image check. If it has the same image version as the WLC, then it proceeds with the next step. If it does not, then it downloads the image from the WLC and reboots to load the new image. In such case, it repeats the process from step 1.
7. Access Point sends a **Configuration Status Request.**
8. WLC sends a **Configuration Status Response**
9. Access Point goes to **RUN** State
10. During the **RUN** state, **CAPWAP Tunnel Maintenance** is performed in two ways:
    1. **Keepalives** are exchanged to maintain the **CAPWAP Data** tunnel
    2. **AP** sends an **Echo Request** to the **WLC**, which has to be answered with its respective **Echo Response**. This is to maintain the **CAPWAP Control** tunnel.


DHCP Server
AP
WLC
DHCP
Discovery Request
Discovery Response
DTLS Session Establishment
DTLS
Join Request
Join Response
Image Data
Configuration Status Request
Configuration Status Response
Change State Event Request
Change State Event Response
RUN
Keepalives
🟥 CAPWAP Control
■ CAPWAP Data
DTLS
Echo Request
Echo Response
Configuration Update Request
Configuration Update Response

CAPWAP Session Establishment Process

Note: As per RFC 5415, CAPWAP uses the UDP Ports 5246 (for CAPWAP Control) and 5247 (for CAPWAP Data).

DTLS Session Establishment

Once the Access Point receives a valid Discovery Response from the WLC, a DTLS tunnel is established between them to transmit all of the subsequent packets over a secured tunnel. This is the process to establish the **DTLS** session:

1. **AP** sends a **Client Hello** message
2. **WLC** sends a **HelloVerifyRequest** message with a cookie used for validation.
3. **AP** sends a **ClientHello** message with a cookie used for validation.
4. **WLC** sends these packets in order:
    1. ServerHello
    2. Certificate
    3. Server Key Exchange
    4. Certificate Request
    5. ServerHelloDone
5. **AP** sends these packets in order:

   1. Certificate
   2. ClientKeyExchange
   3. Certificate Verify
   4. ChangeCipherSpec
6. **WLC** responds to the APs **ChangeCipherSpec** with its own **ChangedCipherSpec**:
   1. **ChangeCipherSpec**

After the last **ChangedCipherSpec** message sent by the WLC, the secure tunnel is established and all traffic sent in both directions is now encrypted.

Wireless LAN Controller Discovery Methods

There are several options to let the Access Points know of the existence of one WLC in the network:

- **DHCP Option 43:** This option provides the APs the IPv4 address of the WLC to join. This process is convenient for large deployments in which the APs and the WLC are in different sites.
- **DHCP Option 52:** This option provides the APs the IPv6 address of the WLC to join. Its usage is convenient in the same scenario as DHCP Option 43.
- **DNS Discovery**: APs queries the domain name **CISCO-CAPWAP-CONTROLLER.localdomain.** You must configure your DNS server to resolve either the IPv4 or IPv6 address of the WLC tto join. This option is convenient for deployments in which the WLCs are stored in the same site as the APs.
- **Layer 3 Broadcast**: The APs automatically send a broadcast message to 255.255.255.255. Any WLC within the same subnet as the AP is expected to respond to this discovery request.
- **Static configuration:** You can use the **capwap ap primary-base <wlc-hostname> <wlc-IP-address>** command to configure a static entry for a WLC in the AP.
- **Mobility Discovery**: If the AP was previously joined to a WLC that was part of a mobility group, the AP also saves a record of the WLCs present in that mobility group.

Note: The WLC Discovery methods listed do not have any precedence order.

Wireless LAN Controller Election

Once the AP has received a **Discovery Response** from any WLC using any of the WLC discovery methods, it selects one controller to join with this criteria:

1. Primary Controller (Configured with the **capwap ap primary-base <wlc-hostname> <wlc-IP-address>** command)
2. Secondary Controller (Configured with the **capwap ap secondary-base <wlc-hostname> <wlc-IP-address>** command)
3. Tertiary Controller (Configured with the **capwap ap tertiary-base <wlc-hostname> <wlc-IP-address>** command**)**
4. If no Primary, Secondary or Tertiary WLC were previously configured, then the AP attempts to join the first WLC that responded to the **Discovery Request** with its own **Discovery Response** that has the maximum capacity of available **APs** (that is, the **WLC** that can support the most **APs** at a given time).

CAPWAP State Machine

In the AP console you can keep track of the CAPWAP State machine, which goes over the steps described in the section CAPWAP Session Establishment.

CAPWAP State: Discovery

Here, you can see the **Discovery Requests** and responses. Observe how the AP does receive a WLC IP via **DHCP** (Option 43), and also sends a **Discovery Request** to previously known WLCs:

<#root>

```
[*09/14/2023 04:12:09.7740]

CAPWAP State: Init


[*09/14/2023 04:12:09.7770]
[*09/14/2023 04:12:09.7770]

CAPWAP State: Discovery


[*09/14/2023 04:12:09.7790]

Discovery Request sent to 172.16.0.20, discovery type STATIC_CONFIG(1)


[*09/14/2023 04:12:09.7800]

Discovery Request

 sent to 172.16.5.11, discovery type STATIC_CONFIG(1)
[*09/14/2023 04:12:09.7800]

Got WLC address 172.16.5.11 from DHCP.


[*09/14/2023 04:12:09.7820]

Discovery Request

 sent to 172.16.0.20, discovery type STATIC_CONFIG(1)
[*09/14/2023 04:12:09.7830]

Discovery Request

 sent to 172.16.5.11, discovery type STATIC_CONFIG(1)
[*09/14/2023 04:12:09.7840]

Discovery Request sent to 255.255.255.255, discovery type UNKNOWN(0)


[*09/14/2023 04:12:09.7850]
[*09/14/2023 04:12:09.7850]

CAPWAP State: Discovery


[*09/14/2023 04:12:09.7850]

Discovery Response

 from 172.16.0.20
[*09/14/2023 04:12:09.8030]

Discovery Response
```

```
 from 172.16.5.11
[*09/14/2023 04:12:09.8060]

Discovery Response

 from 172.16.0.20
[*09/14/2023 04:12:09.8060]

Discovery Response

 from 172.16.5.11
[*09/14/2023 04:12:09.8060]

Discovery Response

 from 172.16.5.11
[*09/14/2023 04:12:09.8060]

Discovery Response

 from 172.16.0.20
[*09/14/2023 04:12:09.8060]

Discovery Response

 from 172.16.5.169
[*09/14/2023 04:12:09.8060]

Discovery Response

 from 172.16.5.169
```

As well as receiving a **Discovery Response** from both a statically configured WLC (172.16.0.20) and the WLC indicated via DHCP Option 43 (172.16.5.11), this AP also received a **Discovery Response** from another WLC (172.16.5.169) within the same subnet because it got the broadcast Discovery message.

CAPWAP State: DTLS Setup.

Here, the DTLS session between the AP and the WLC is exchanged.

<#root>

```
[*09/27/2023 21:50:41.0000]

CAPWAP State: DTLS Setup


[*09/27/2023 21:50:41.7140] sudi99_request_check_and_load: Use HARSA SUDI certificat
```

CAPWAP State: Join

After establishing the DTLS session, a **Join Request** to the WLC is now sent over the secure session. Observe how this request is immediately answered with a **Join Response** from the WLC

<#root>

```
[*09/27/2023 21:50:41.9880]
```

```
CAPWAP State: Join


[*09/27/2023 21:50:41.9910]

Sending Join request to 172.16.5.11

 through port 5270
[*09/27/2023 21:50:41.9950]

Join Response from 172.16.5.11


[*09/27/2023 21:50:41.9950]

AC accepted join request

 with result code: 0
[*09/27/2023 21:50:41.9990] Received wlcType 0, timer 30
[*09/27/2023 21:50:41.9990] TLV ID 2216 not found
[*09/27/2023 21:50:41.9990] TLV-DEC-ERR-1: No proc for 2216
```

CAPWAP State: Image Data

The AP compares its image with the WLC image. In this case, both the APs active partition and its backup partition have different images than the WLC, so it invokes the **upgrade.sh** script, which instructs the AP to request the adequate image to the WLC and download it into its current non-active partition.

<#root>

```
[*09/27/2023 21:50:42.0430]

CAPWAP State: Image Data


[*09/27/2023 21:50:42.0430]

AP image version 8.10.185.0 backup 8.10.105.0, Controller 17.9.3.50


[*09/27/2023 21:50:42.0430]

Version does not match.


[*09/27/2023 21:50:42.0680]

upgrade.sh

: Script called with args:[PRECHECK]
[*09/27/2023 21:50:42.1060] do PRECHECK,

part2 is active part


[*09/27/2023 21:50:42.1240]

upgrade.sh

: /tmp space: OK available 101476, required 40000
[*09/27/2023 21:50:42.1250] wtpImgFileReadRequest: request ap1g7, local /tmp/part.tar
[*09/27/2023 21:50:42.1310]
```

Image Data Request sent to 172.16.5.11

```
, fileName [ap1g7], slaveStatus 0
[*09/27/2023 21:50:42.1340]
```

Image Data Response from 172.16.5.11

```
[*09/27/2023 21:50:42.1340] AC accepted join request with result code: 0
[*09/27/2023 21:50:42.1450] <.................................................
[*09/27/2023 21:50:55.4980] .................................................
[*09/27/2023 21:51:11.6290] ..............................Discarding msg CAPWAP_WTP_EVENT_REQUEST(type
[*09/27/2023 21:51:19.7220] ..................
[*09/27/2023 21:51:24.6880] .................................................
[*09/27/2023 21:51:37.7790] .................................................
[*09/27/2023 21:51:50.9440] ..................................> 76738560 bytes, 57055 msgs, 930 last
[*09/27/2023 21:51:59.9160] Last block stored, IsPre 0, WriteTaskId 0
[*09/27/2023 21:51:59.9160]
```

Image transfer completed from WLC

```
, last 1
```

Once the image transfer is completed, the AP initiates an image signing verification process to validate it. After doing so, the **upgrade.sh** script installs the image into the current non-active partition, and swaps the partition it boots from. Finally, the AP reloads itself and repeats the process from the beginning (**CAPWAP State: Discover**).

<#root>

```
[*09/27/2023 21:52:01.1280]
```

Image signing verify success.

```
[*09/27/2023 21:52:01.1440]
[*09/27/2023 21:52:01.1440] [9/27/2023 21:53:2] : Shadow is now in-synced with master
[*09/27/2023 21:52:01.1440]
[*09/27/2023 21:52:01.1440] [9/27/2023 21:53:2] : Verifying against bundle image btldr.img...
[*09/27/2023 21:52:01.1570]
```

upgrade.sh

```
:
```

part to upgrade is part1

```
[*09/27/2023 21:52:01.1780]
```

upgrade.sh

```
: AP version1: part1 8.10.105.0, img 17.9.3.50
[*09/27/2023 21:52:01.1960]
```

upgrade.sh

```
: Extracting and verifying image in part1...
[*09/27/2023 21:52:01.2080]
```

upgrade.sh

```
: BOARD generic case execute
```

```
[*09/27/2023 21:52:01.5280]

upgrade.sh

: Untar /tmp/part.tar to /bootpart/part1...
[*09/27/2023 21:52:01.7890]

upgrade.sh

: Sync image to disk...
[*09/27/2023 21:52:31.4970]

upgrade.sh

: status '

Successfully verified image in part1.

'
[*09/27/2023 21:52:32.5270]

upgrade.sh

: AP version2: part1 17.9.3.50, img 17.9.3.50
[*09/27/2023 21:52:32.5540]

upgrade.sh

: AP backup version: 17.9.3.50
[*09/27/2023 21:52:32.5700]

upgrade.sh

:

Finished upgrade task.


[*09/27/2023 21:52:32.5840]

upgrade.sh

: Cleanup for do_upgrade...
[*09/27/2023 21:52:32.5970]

upgrade.sh

: /tmp/upgrade_in_progress cleaned
[*09/27/2023 21:52:32.6090]

upgrade.sh

: Cleanup tmp files ...
[*09/27/2023 21:52:32.6720]

upgrade.sh

: Script called with args:[ACTIVATE]
[*09/27/2023 21:52:32.7100] do ACTIVATE, part2 is active part
[*09/27/2023 21:52:32.7640]

upgrade.sh

: Verifying image signature in part1
[*09/27/2023 21:52:33.7730]

upgrade.sh

: status 'Successfully verified image in part1.'
```

```
[*09/27/2023 21:52:33.7850]

upgrade.sh

:

activate part1, set BOOT to part1


[*09/27/2023 21:52:34.2940]

upgrade.sh

:

AP primary version after reload: 17.9.3.50


[*09/27/2023 21:52:34.3070]

upgrade.sh

: AP backup version after reload: 8.10.185.0
[*09/27/2023 21:52:34.3190]

upgrade.sh

: Create after-upgrade.log
[*09/27/2023 21:52:37.3520]

AP Rebooting: Reset Reason - Image Upgrade
```

Warning: Wave 1 Access Points could fail to download a new image due to an expired certificate. Please refer to Field Notice 72524 for more information and read carefully the Cisco IOS AP Image Download Fails Due to Expired Image Signing Certificate Past December 4th, 2022 (CSCwd80290) Support Document to understand its impact and solution.

Once the AP reloads and goes again through the **CAPWAP Discover** and **Join** states, during the **Image Data** state it detects that it now has the adequate image.

<#root>

```
[*09/27/2023 21:56:13.7640]

CAPWAP State: Image Data


[*09/27/2023 21:56:13.7650]

AP image version 17.9.3.50 backup 8.10.185.0, Controller 17.9.3.50


[*09/27/2023 21:56:13.7650]
```

```
Version is the same, do not need update.


[*09/27/2023 21:56:13.7650] status '

upgrade.sh: Script called with args:[NO_UPGRADE]

'
[*09/27/2023 21:56:13.7850] do NO_UPGRADE, part1 is active part
```

CAPWAP State: Configure

After the AP validates that it has the same version as the WLC, it notifies its current configurations to the WLC. In general, this means that the AP asks to maintain its configurations (if they are available in the WLC).

<#root>

```
[*09/27/2023 21:56:14.8680]

CAPWAP State: Configure


[*09/27/2023 21:56:15.8890] Telnet is not supported by AP, should not encode this payload
[*09/27/2023 21:56:15.8890] Radio [1] Administrative state DISABLED change to ENABLED
[*09/27/2023 21:56:16.0650] Radio [0] Administrative state DISABLED change to ENABLED
[*09/27/2023 21:56:16.0750] DOT11_CFG[1]: Starting radio 1
[*09/27/2023 21:56:16.1150] DOT11_DRV[1]: Start Radio1
[*09/27/2023 21:56:16.1160] DOT11_DRV[1]: set_channel Channel set to 36/20
[*09/27/2023 21:56:16.4380] Started Radio 1
[*09/27/2023 21:56:16.4880] DOT11_CFG[0]: Starting radio 0
[*09/27/2023 21:56:17.5220] DOT11_DRV[0]: Start Radio0
[*09/27/2023 21:56:16.5650] DOT11_DRV[0]: set_channel Channel set to 1/20
[*09/27/2023 21:56:16.5650] Started Radio 0
[*09/27/2023 21:56:16.5890] sensord psage_base init: RHB Sage base ptr a1030000
```

CAPWAP State: Run

At this point, the AP has successfully joined the controller. During this state, the WLC triggers a mechanism to override the configuration requested by the AP. You can see that the AP gets **Radio and Credentials configurations** pushed, and it also gets assigned to the **default policy tag** since the WLC had no previous knowledge of this AP.

<#root>

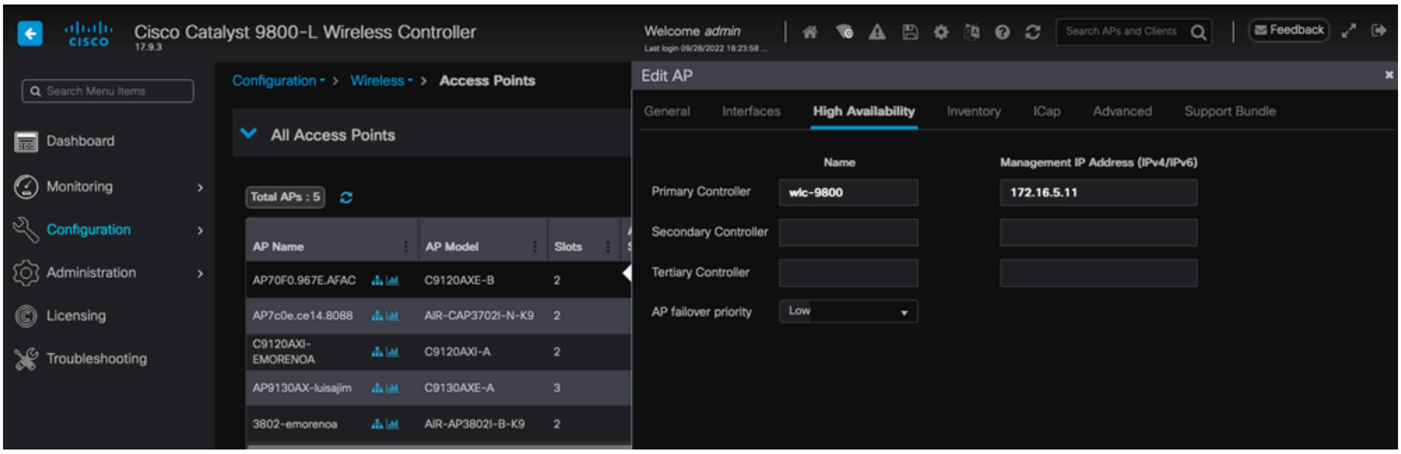
```
[*09/27/2023 21:56:17.4870]

CAPWAP State: Run


[*09/27/2023 21:56:17.4870]

AP has joined controller

 uwu-9800
[*09/27/2023 21:56:17.4940] DOT11_DRV[0]: set_channel Channel set to 1/20
```

```
[*09/27/2023 21:56:17.5440] sensord split_glue psage_base: RHB Sage base ptr a1030000
[*09/27/2023 21:56:17.6010] sensord split_glue sage_addr: RHB Sage base ptr a1030000
[*09/27/2023 21:56:17.6230] ptr a1030000
[*09/27/2023 21:56:17.6420]

DOT11_DRV[0]: set_channel Channel set to 1/20


[*09/27/2023 21:56:17.8120]

DOT11_DRV[1]: set_channel Channel set to 36/20


[*09/27/2023 21:56:17.9350] Previous AP mode is 0, change to 0
[*09/27/2023 21:56:18.0160] Current session mode: ssh, Configured: Telnet-No, SSH-Yes, Console-Yes
[*09/27/2023 21:56:18.1220] Current session mode: telnet, Configured: Telnet-No, SSH-Yes, Console-Yes
[*09/27/2023 21:56:18.1310] Current session mode: console, Configured: Telnet-No, SSH-Yes, Console-Yes
[*09/27/2023 21:56:18.1340]

chpasswd: password for user changed


[*09/27/2023 21:56:18.1350]

chpasswd: password for user changed


[*09/27/2023 21:56:18.1520] systemd[1]: Starting Cisco rsyslog client watcher...
[*09/27/2023 21:56:18.1610] Same LSC mode, no action needed
[*09/27/2023 21:56:18.1640] CLSM[00:00:00:00:00:00]: U3 Client RSSI Stats feature is deprecated; can no
[*09/27/2023 21:56:18.1720] systemd[1]: Stopping rsyslog client...
[*09/27/2023 21:56:18.2120] systemd[1]: Starting Cisco syslog service...
[*09/27/2023 21:56:18.2230] systemd[1]: Started Cisco syslog service.
[*09/27/2023 21:56:18.2410] systemd[1]: Started rsyslog client.
[*09/27/2023 21:56:18.2440] AP is in good condition, BLE is off
[*09/27/2023 21:56:18.2510] SET_SYS_COND_INTF: allow_usb state: 1 (up) condition
[*09/27/2023 21:56:18.2530] systemd[1]: Starting dhcpv6 client watcher...
[*09/27/2023 21:56:18.2530] systemd[1]: Stopping DHCPv6 client...
[*09/27/2023 21:56:18.2530] systemd[1]: Starting DHCPv6 client...
[*09/27/2023 21:56:18.2530] systemd[1]: Started DHCPv6 client.
[*09/27/2023 21:56:18.2530] systemd[1]: Started dhcpv6 client watcher.
[*09/27/2023 21:56:18.2560]

Set radio 0 power 4 antenna mask 15


[*09/27/2023 21:56:18.2530]

Set radio 1 power 4 antenna mask 15


[*09/27/2023 21:56:18.2530] Got WSA Server config TLVs
[*09/27/2023 21:56:18.2720]

AP tag change to default-policy-tag


[*09/27/2023 21:56:18.2780] Chip flash OK
```

Configure

Static WLC Election

In the GUI, you can go to **Configuration > Wireless > Access Points**, select an AP and navigate to the **High Availability** tab. Here, you can configure the **Primary, Secondary and Tertiary** WLCs, as described in the Wireless LAN Controller Election section of this document. This configuration is done per Access Point.



AP Name	AP Model	Slots
AP70F0.967E.AFAC	C9120AXE-B	2
AP7c0e.ce14.8088	AIR-CAP3702I-N-K9	2
C9120AXI-EMORENOA	C9120AXI-A	2
AP9130AX-luisajim	C9130AXE-A	3
3802-emorenoa	AIR-AP3802I-B-K9	2

	Name	Management IP Address (IPv4/IPv6)
Primary Controller	wlc-9800	172.16.5.11
Secondary Controller		
Tertiary Controller		

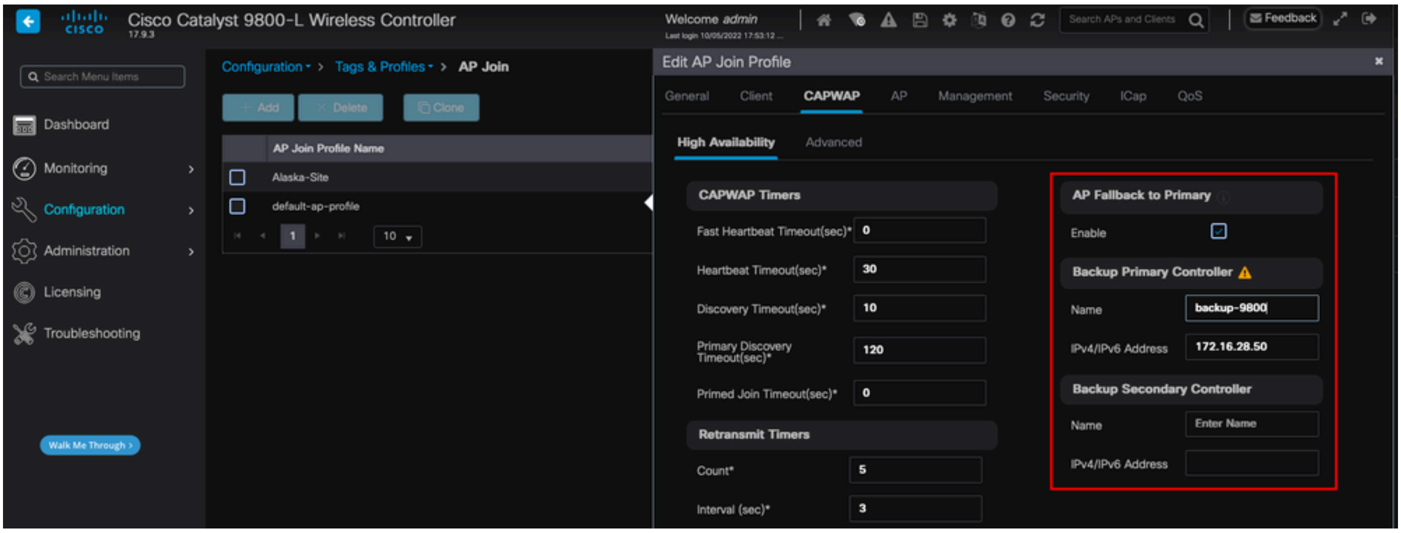
AP failover priority: Low

Primary, Secondary and Tertiary WLCs for an AP.

Note: Starting Cisco IOS XE 17.9.2, you can use Priming Profiles to configure primary, secondary, and tertiary controllers for a group of APs matching regular expression (regex) or for an individual AP. Refer to the AP Fallback to Controllers Configured Under AP Priming Profile section of the Configuration Guide for further information.

Please note that the **Primary, Secondary and Tertiary Controllers** configured in the AP **High Availability** tab differ from the **Backup Primary and Secondary** WLCs that can be configured per **AP Join Profile** under the **CAPWAP > High Availability** tab**.** The **Primary, Secondary and Tertiary Controllers** are considered WLCs with priorities 1, 2 and 3, respectively, while the **Backup Primary and Secondary** are considered WLCs with priorities 4 and 5.

If **AP Fallback** is enabled, the AP actively looks for the **Primary Controller** when joined to a different **WLC**. The **AP** only looks for **WLCs** with priorities 4 and 5 once there is a **CAPWAP Down** event and none of the **Backup Primary and Secondary Controllers** are available.



CAPWAP Timers

Fast Heartbeat Timeout(sec)* 0

Heartbeat Timeout(sec)* 30

Discovery Timeout(sec)* 10

Primary Discovery Timeout(sec)* 120

Primed Join Timeout(sec)* 0

Retransmit Timers

Count* 5

Interval (sec)* 3

AP Fallback to Primary

Enable ☑

Backup Primary Controller

Name backup-9800

IPv4/IPv6 Address 172.16.28.50

Backup Secondary Controller

Name Enter Name

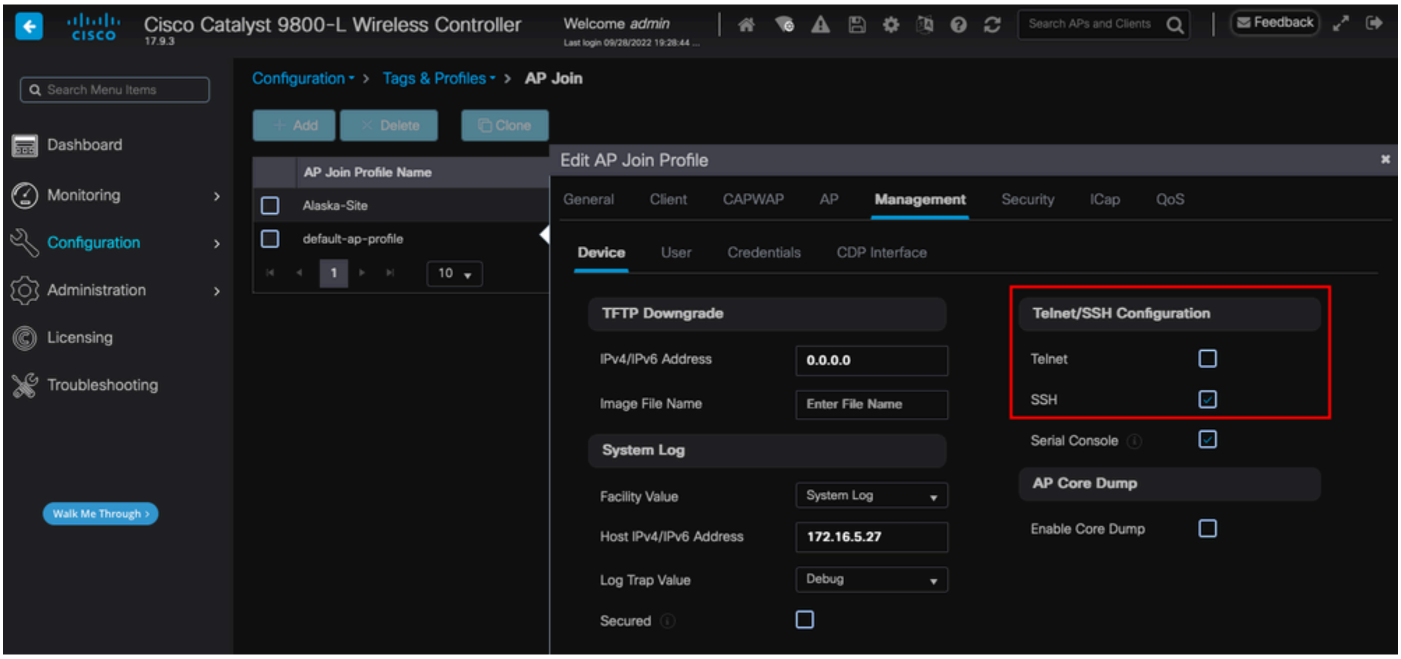
IPv4/IPv6 Address

High Availability Options in the AP Join Profile

Note: The configuration of **Backup Primary** and **Backup Secondary** WLCs in the **AP Join Profile** does not populate the **Static Primary** and **Secondary** entries in the **High Availability** Tab of the Access Point.

Enabling Telnet/SSH Access to the AP

Go to **Configuration > Tags & Profiles > AP Join > Management > Device** and select **SSH** and/or **Telnet**.



TFTP Downgrade

IPv4/IPv6 Address: 0.0.0.0

Image File Name: Enter File Name

System Log

Facility Value: System Log

Host IPv4/IPv6 Address: 172.16.5.27

Log Trap Value: Debug

Secured ☐

Telnet/SSH Configuration

Telnet ☐

SSH ☑

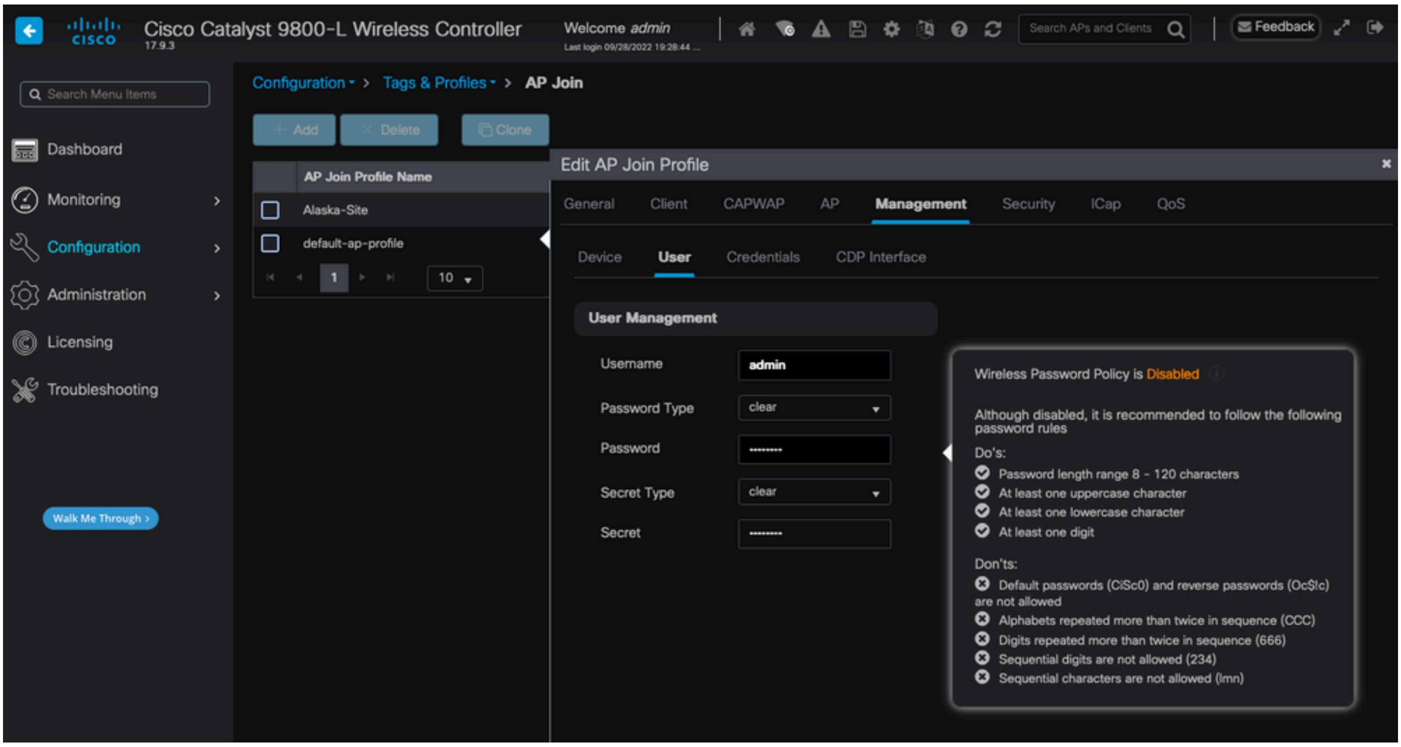
Serial Console ☑

AP Core Dump

Enable Core Dump ☐

Enable Telnet/SSH Access on the AP Join Profile

To configure the SSH/Telnet Credentials, navigate to the **User** tab in the same window and set the **Username, Password** and **Secret** to access the AP.



User Management

Username: admin

Password Type: clear

Password: ••••••••

Secret Type: clear

Secret: ••••••••

Wireless Password Policy is Disabled

Although disabled, it is recommended to follow the following password rules

Do's:

- Password length range 8 - 120 characters
- At least one uppercase character
- At least one lowercase character
- At least one digit

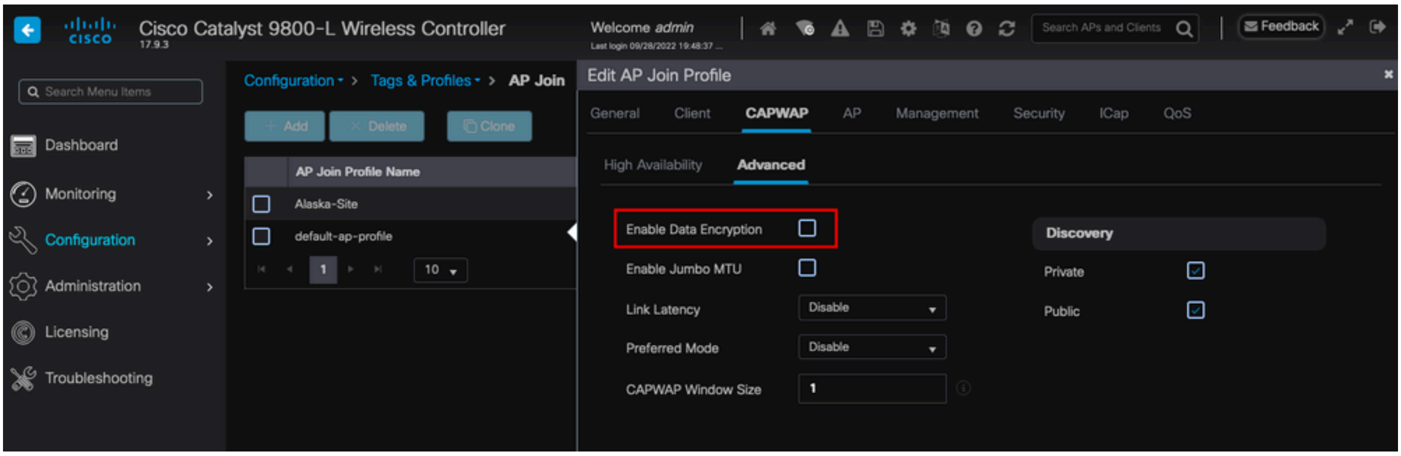
Don'ts:

- Default passwords (CiScO) and reverse passwords (Oc$Ic) are not allowed
- Alphabets repeated more than twice in sequence (CCC)
- Digits repeated more than twice in sequence (666)
- Sequential digits are not allowed (234)
- Sequential characters are not allowed (lmn)

SSH and Telnet Credentials for the AP

Data Link Encryption

If you need to troubleshoot any client issue that requires you to take a packet capture of the APs traffic, ensure that **Data Link Encryption** is not enabled under **Configuration > Tags & Profiles > AP Join > CAPWAP > Advanced**. Otherwise, your traffic comes encrypted.



Enable Data Encryption □

Enable Jumbo MTU □

Link Latency: Disable

Preferred Mode: Disable

CAPWAP Window Size: 1

Private ☑

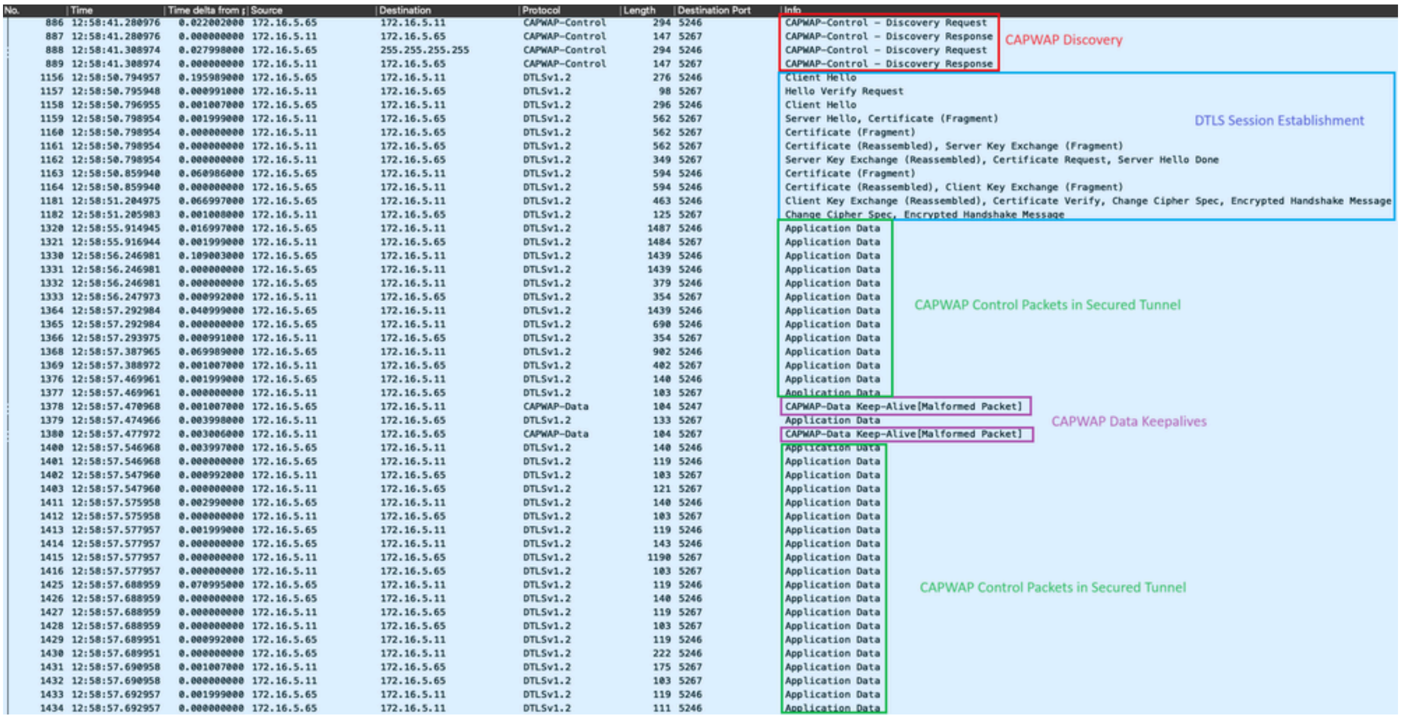
Public ☑

Data Link Encryption

Note: Data Encryption only encrypts CAPWAP Data traffic. CAPWAP Control traffic is already encrypted via DTLS.

Verify

In addition to tracking the CAPWAP state machine in the APs console, you can also take an Embedded Packet Capture in the WLC to analyze the AP Join process:



```
No.   Time             Time delta from  Source       Destination      Protocol        Length Destination Port  Info
  886 12:58:41.280976  0.022002000  172.16.5.65  172.16.5.11      CAPWAP-Control     294 5246  CAPWAP-Control - Discovery Request
  887 12:58:41.280976  0.000000000  172.16.5.11  172.16.5.65      CAPWAP-Control     147 5267  CAPWAP-Control - Discovery Response
  888 12:58:41.308974  0.027998000  172.16.5.65  255.255.255.255  CAPWAP-Control     294 5246  CAPWAP-Control - Discovery Request
  889 12:58:41.308974  0.000000000  172.16.5.11  172.16.5.65      CAPWAP-Control     147 5267  CAPWAP-Control - Discovery Response
 1156 12:58:50.794957  0.195989000  172.16.5.65  172.16.5.11      DTLSv1.2           276 5246  Client Hello
 1157 12:58:50.795948  0.000991000  172.16.5.11  172.16.5.65      DTLSv1.2            98 5267  Hello Verify Request
 1158 12:58:50.796955  0.001007000  172.16.5.65  172.16.5.11      DTLSv1.2           296 5246  Client Hello
 1159 12:58:50.798954  0.001999000  172.16.5.11  172.16.5.65      DTLSv1.2           562 5267  Server Hello, Certificate (Fragment)
 1160 12:58:50.798954  0.000000000  172.16.5.11  172.16.5.65      DTLSv1.2           562 5267  Certificate (Fragment)
 1161 12:58:50.798954  0.000000000  172.16.5.11  172.16.5.65      DTLSv1.2           562 5267  Certificate (Reassembled), Server Key Exchange (Fragment)
 1162 12:58:50.798954  0.000000000  172.16.5.11  172.16.5.65      DTLSv1.2           349 5267  Server Key Exchange (Reassembled), Certificate Request, Server Hello Done
 1163 12:58:50.859940  0.060986000  172.16.5.65  172.16.5.11      DTLSv1.2           594 5246  Certificate (Fragment)
 1164 12:58:50.859940  0.000000000  172.16.5.65  172.16.5.11      DTLSv1.2           594 5246  Certificate (Reassembled), Client Key Exchange (Fragment)
 1181 12:58:51.204975  0.066997000  172.16.5.65  172.16.5.11      DTLSv1.2           463 5246  Client Key Exchange (Reassembled), Certificate Verify, Change Cipher Spec, Encrypted Handshake Message
 1182 12:58:51.205983  0.001008000  172.16.5.11  172.16.5.65      DTLSv1.2           125 5267  Change Cipher Spec, Encrypted Handshake Message
 1320 12:58:55.914945  0.016997000  172.16.5.65  172.16.5.11      DTLSv1.2          1487 5246  Application Data
 1321 12:58:55.916944  0.001999000  172.16.5.11  172.16.5.65      DTLSv1.2          1484 5267  Application Data
 1330 12:58:56.246981  0.109003000  172.16.5.65  172.16.5.11      DTLSv1.2          1439 5246  Application Data
 1331 12:58:56.246981  0.000000000  172.16.5.65  172.16.5.11      DTLSv1.2          1439 5246  Application Data
 1332 12:58:56.246981  0.000000000  172.16.5.65  172.16.5.11      DTLSv1.2           379 5246  Application Data
 1333 12:58:56.247973  0.000992000  172.16.5.11  172.16.5.65      DTLSv1.2           354 5267  Application Data
 1364 12:58:57.292984  0.040999000  172.16.5.65  172.16.5.11      DTLSv1.2          1439 5246  Application Data
 1365 12:58:57.292984  0.000000000  172.16.5.65  172.16.5.11      DTLSv1.2           690 5246  Application Data
 1366 12:58:57.293975  0.000991000  172.16.5.11  172.16.5.65      DTLSv1.2           354 5267  Application Data
 1368 12:58:57.387965  0.069989000  172.16.5.65  172.16.5.11      DTLSv1.2           902 5246  Application Data
 1369 12:58:57.388972  0.001007000  172.16.5.11  172.16.5.65      DTLSv1.2           402 5267  Application Data
 1376 12:58:57.469961  0.001999000  172.16.5.65  172.16.5.11      DTLSv1.2           140 5246  Application Data
 1377 12:58:57.469961  0.000000000  172.16.5.11  172.16.5.65      DTLSv1.2           103 5267  Application Data
 1378 12:58:57.470968  0.001007000  172.16.5.65  172.16.5.11      CAPWAP-Data        104 5247  CAPWAP-Data Keep-Alive[Malformed Packet]
 1379 12:58:57.474966  0.003998000  172.16.5.11  172.16.5.65      DTLSv1.2           133 5267  Application Data
 1380 12:58:57.477972  0.003006000  172.16.5.11  172.16.5.65      CAPWAP-Data        104 5267  CAPWAP-Data Keep-Alive[Malformed Packet]
 1400 12:58:57.546968  0.003997000  172.16.5.65  172.16.5.11      DTLSv1.2           140 5246  Application Data
 1401 12:58:57.546968  0.000000000  172.16.5.65  172.16.5.11      DTLSv1.2           119 5246  Application Data
 1402 12:58:57.547960  0.000992000  172.16.5.11  172.16.5.65      DTLSv1.2           103 5267  Application Data
 1403 12:58:57.547960  0.000000000  172.16.5.11  172.16.5.65      DTLSv1.2           121 5267  Application Data
 1411 12:58:57.575958  0.002990000  172.16.5.65  172.16.5.11      DTLSv1.2           140 5246  Application Data
 1412 12:58:57.575958  0.000000000  172.16.5.11  172.16.5.65      DTLSv1.2           103 5267  Application Data
 1413 12:58:57.577957  0.001999000  172.16.5.65  172.16.5.11      DTLSv1.2           119 5246  Application Data
 1414 12:58:57.577957  0.000000000  172.16.5.65  172.16.5.11      DTLSv1.2           143 5246  Application Data
 1415 12:58:57.577957  0.000000000  172.16.5.11  172.16.5.65      DTLSv1.2          1190 5267  Application Data
 1416 12:58:57.577957  0.000000000  172.16.5.11  172.16.5.65      DTLSv1.2           103 5267  Application Data
 1425 12:58:57.688959  0.070995000  172.16.5.65  172.16.5.11      DTLSv1.2           119 5246  Application Data
 1426 12:58:57.688959  0.000000000  172.16.5.65  172.16.5.11      DTLSv1.2           140 5246  Application Data
 1427 12:58:57.688959  0.000000000  172.16.5.11  172.16.5.65      DTLSv1.2           119 5267  Application Data
 1428 12:58:57.688959  0.000000000  172.16.5.11  172.16.5.65      DTLSv1.2           103 5267  Application Data
 1429 12:58:57.689951  0.000992000  172.16.5.65  172.16.5.11      DTLSv1.2           119 5246  Application Data
 1430 12:58:57.689951  0.000000000  172.16.5.65  172.16.5.11      DTLSv1.2           222 5246  Application Data
 1431 12:58:57.690958  0.001007000  172.16.5.11  172.16.5.65      DTLSv1.2           175 5267  Application Data
 1432 12:58:57.690958  0.000000000  172.16.5.11  172.16.5.65      DTLSv1.2           103 5267  Application Data
 1433 12:58:57.692957  0.001999000  172.16.5.65  172.16.5.11      DTLSv1.2           119 5246  Application Data
 1434 12:58:57.692957  0.000000000  172.16.5.65  172.16.5.11      DTLSv1.2           111 5246  Application Data
```

CAPWAP Discovery
DTLS Session Establishment
CAPWAP Control Packets in Secured Tunnel
CAPWAP Data Keepalives
CAPWAP Control Packets in Secured Tunnel

AP Join Process seen in an Embedded Packet Capture in the WLC

Note how all traffic after the **Chance Cipher Spec** packet (Packet No. 1182) is shown only as **Application Data** over **DTLSv1.2**. This is all the encrypted data after the **DTLS session establishment**.

Troubleshoot

Known Issues

Please refer to the known issues that could prevent your APs to join the WLC.

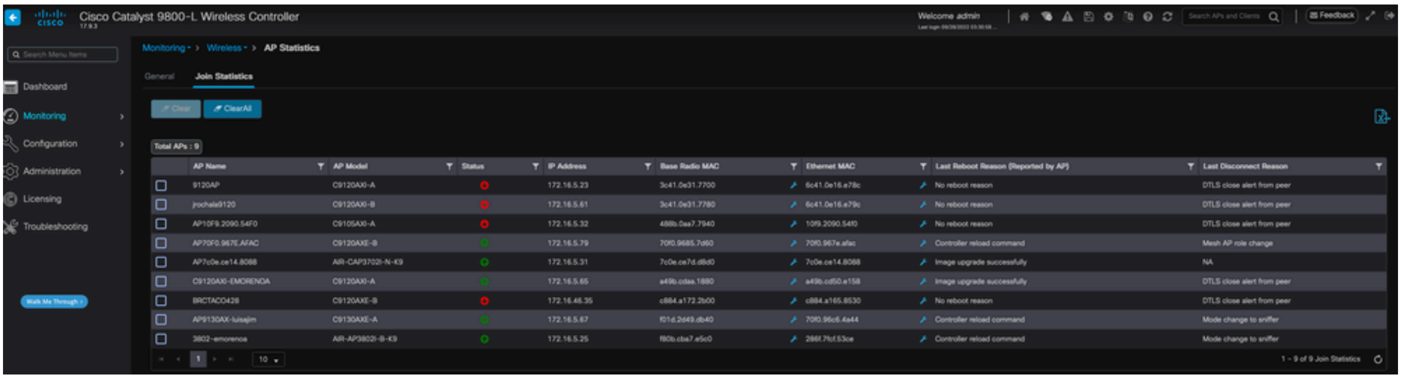
- APs on boot loop due to corrupted image in Wave 2 and Catalyst 11ax Access Points (CSCvx32806)
- Field Notice 72424: C9105/C9120/C9130 Access Points manufactured starting in September 2022 may require software upgrades to join Wireless LAN Controllers.
- Field Notice 72524: During Software Upgrade/Downgrade, Cisco IOS APs Might Remain in Downloading State After December 4, 2022 Due to Certificate Expiration - Software Upgrade Recommended
- Cisco bug ID CSCwb13784: APs not able to join 9800 due to invalid path MTU in AP Join request
- Cisco bug ID CSCvu22886: C9130: message "unlzma: write: No space left on device" on upgrade to 17.7 Increase max size of /tmp

Always refer to the **Upgrade Path** section of the Release Notes of each version before upgrading.

Note: Starting from Cisco IOS XE Cupertino 17.7.1, the Cisco Catalyst 9800-CL Wireless Controller does not accept more than 50 APs if the smart licensing is not connected and up.

WLC GUI Checks

On your WLC, go to **Monitoring > Wireless > AP Statistics > Join Statistics** you can see the **Last Reboot Reason** reported by any AP and the **Last Disconnect Reason** registered by the WLC.



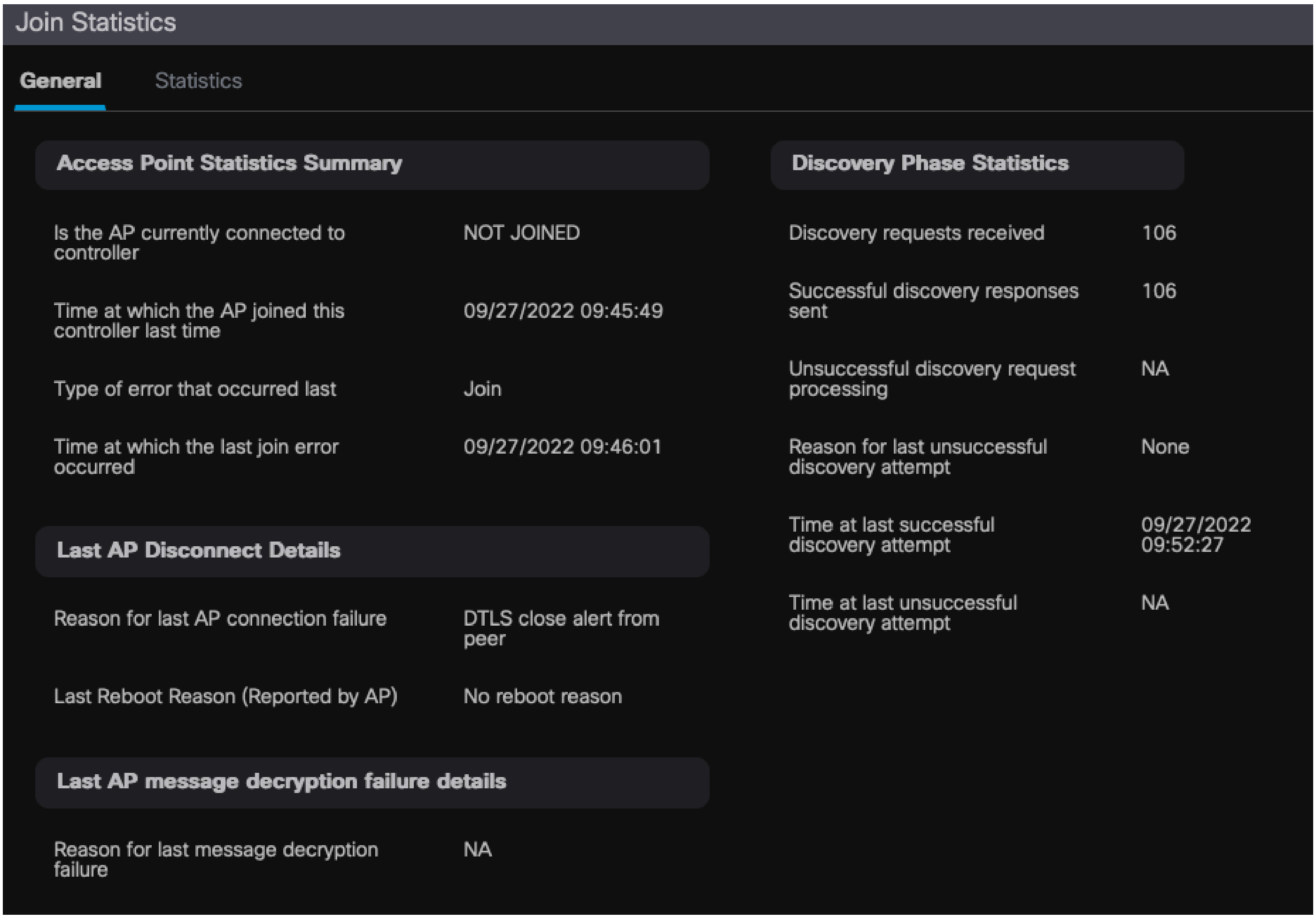
Total APs : 9

	AP Name	AP Model	Status	IP Address	Base Radio MAC	Ethernet MAC	Last Reboot Reason (Reported by AP)	Last Disconnect Reason
□	9120AP	C9120AXI-A	🔴	172.16.5.23	3c41.0e31.7700	6c41.0e16.e78c	No reboot reason	DTLS close alert from peer
□	jrochala9120	C9120AXI-B	🔴	172.16.5.61	3c41.0e31.7780	6c41.0e16.e79c	No reboot reason	DTLS close alert from peer
□	AP10F9.2090.54F0	C9105AXI-A	🔴	172.16.5.32	488b.0aa7.7940	10f9.2090.54f0	No reboot reason	DTLS close alert from peer
□	AP70F0.967E.AFAC	C9120AXE-B	🟢	172.16.5.79	70f0.9685.7d60	70f0.967e.afac	Controller reload command	Mesh AP role change
□	AP7c0e.ce14.8088	AIR-CAP3702I-N-K9	🟢	172.16.5.31	7c0e.ce7d.d8d0	7c0e.ce14.8088	Image upgrade successfully	NA
□	C9120AXI-EMORENOA	C9120AXI-A	🟢	172.16.5.65	a49b.cdaa.1880	a49b.cd50.e158	Image upgrade successfully	DTLS close alert from peer
□	BRCTACO428	C9120AXE-B	🔴	172.16.46.35	c884.a172.2b00	c884.a165.8530	No reboot reason	DTLS close alert from peer
□	AP9130AX-luisajim	C9130AXE-A	🟢	172.16.5.67	f01d.2d49.db40	70f0.96c6.4a44	Controller reload command	Mode change to sniffer
□	3802-emorenoa	AIR-AP3802I-B-K9	🟢	172.16.5.25	f80b.cba7.e5c0	286f.7fcf.53ce	Controller reload command	Mode change to sniffer

1 - 9 of 9 Join Statistics

AP Join Statistics page on the WLC

You can click on any AP and check for AP Join Statistics details. Here, you can see more detailed information, like the time and date at which the AP last joined and attempted to discover the WLC.



Join Statistics

Access Point Statistics Summary

Is the AP currently connected to controller	NOT JOINED
Time at which the AP joined this controller last time	09/27/2022 09:45:49
Type of error that occurred last	Join
Time at which the last join error occurred	09/27/2022 09:46:01

Last AP Disconnect Details

Reason for last AP connection failure	DTLS close alert from peer
Last Reboot Reason (Reported by AP)	No reboot reason

Last AP message decryption failure details

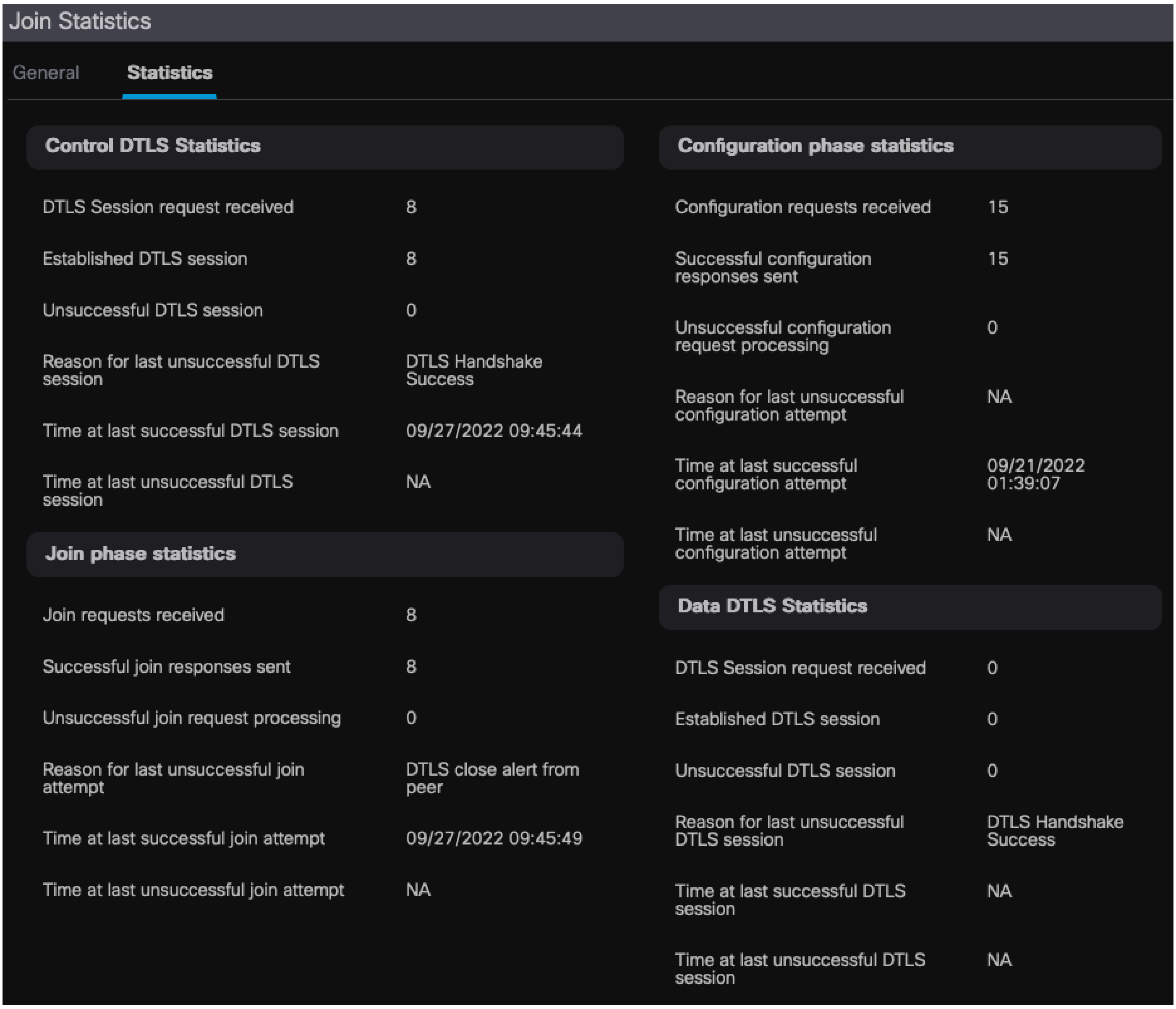
Reason for last message decryption failure	NA

Discovery Phase Statistics

Discovery requests received	106
Successful discovery responses sent	106
Unsuccessful discovery request processing	NA
Reason for last unsuccessful discovery attempt	None
Time at last successful discovery attempt	09/27/2022 09:52:27
Time at last unsuccessful discovery attempt	NA

General AP Join Statistics

For more detailed information, go to the Statistics tab of the same window. Here you can compare the amount of **Join Responses Sent** with the amount of **Join Requests Received**, as well as the **Configuration Responses Sent** versus the **Configuration Requests Received**.



Join Statistics

Control DTLS Statistics

DTLS Session request received	8
Established DTLS session	8
Unsuccessful DTLS session	0
Reason for last unsuccessful DTLS session	DTLS Handshake Success
Time at last successful DTLS session	09/27/2022 09:45:44
Time at last unsuccessful DTLS session	NA

Join phase statistics

Join requests received	8
Successful join responses sent	8
Unsuccessful join request processing	0
Reason for last unsuccessful join attempt	DTLS close alert from peer
Time at last successful join attempt	09/27/2022 09:45:49
Time at last unsuccessful join attempt	NA

Configuration phase statistics

Configuration requests received	15
Successful configuration responses sent	15
Unsuccessful configuration request processing	0
Reason for last unsuccessful configuration attempt	NA
Time at last successful configuration attempt	09/21/2022 01:39:07
Time at last unsuccessful configuration attempt	NA

Data DTLS Statistics

DTLS Session request received	0
Established DTLS session	0
Unsuccessful DTLS session	0
Reason for last unsuccessful DTLS session	DTLS Handshake Success
Time at last successful DTLS session	NA
Time at last unsuccessful DTLS session	NA

Detailed AP Join Statistics

Commands

These commands are useful to troubleshoot AP Join issues:

From the WLC

- **show ap summary**
- **debug capwap error**
- **debug capwap packet**

From Wave 2 and Catalyst 11ax APs

- **debug capwap client events**
- **debug capwap client error**
- **debug dtls client error**
- **debug dtls client event**
- **debug capwap client keepalive**
- **test capwap restart**

- **capwap ap erase all**

From Wave 1 APs

- **debug capwap console cli**
- **debug capwap client no-reload**
- **show dtls stats**
- **clear cawap ap all-config**

Note: When you connect to the APs via Telnet/SSH to troubleshoot, always issue the command **terminal monitor** while reproducing the issue after enabling debugs on the APs. Otherwise, you can not see any output from the debugs.

Radioactive Traces

A good starting point when troubleshooting AP Join issues is to take Radioactive Traces of both the Radio and Ethernet MAC addresses of an AP that has issues joining. Refer to the Debug & Log collection on Catalyst 9800 WLC Document for details on generating these logs.