Troubleshoot COS APs

Contents

Introduction

This document describes some of the troubleshooting tools available for APs running the COS operating system (Cheetah OS, Click OS, simply Cisco AP OS).

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document focuses on COS APs like APs models of the series 2800, 3800, 1560 and 4800, as well as new 11ax APs Catalyst 91xx.

This document focuses on many features available in AireOS 8.8 and later. And also Cisco IOS® XE 16.12.2s and later.

There can be comments about availability of certain features in prior releases.

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.

Capture Packet Traces (Sniffer Traces)

Wired PCAP on AP Port

It is possible (as of 8.7 with the filter available in 8.8) to take a pcap on the AP ethernet port. You can either display the result live on the CLI (with only summarized packet details) or save it as a full pcap in the AP flash.

The wired pcap captures everything on the ethernet side (both Rx/Tx) and the tap point inside the AP is imediately before the packet is put on wire.

However, it only captures AP CPU-plane traffic, which means traffic to and from the AP (AP DHCP, AP capwap control tunnel, ...) and does not show client traffic.

Note that the size is very limited (Max size limit of 5MB), so it can be required to configure filters to capture only the traffic you are interested in.

Ensure to stop the traffic capture with "no debug traffic wired ip capture" or simply "undebug all" before you try to copy it (otherwise the copy does not end as packets are still written).

Procedure

Step 1. Start the pcap; select the traffic type with “debug traffic wired ip capture”:

<#root>

```
AP70DB.98E1.3DEC#debug traffic wired ip capture
% Writing packets to "/tmp/pcap/
```

```
AP70DB.98E1.3DEC_capture.pcap0"
```

```
AP70DB.98E1.3DEC#reading from file /dev/click_wired_log, link-type EN10MB (Ethernet)
```

Step 2. Wait for the traffic to flow and then Stop the capture with the command "no debug traffic wired ip capture" or simply "undebug all":

```
AP70DB.98E1.3DEC#no debug traffic wired ip capture
```

Step 3. Copy the file to tftp/scp server:

<#root>

```
AP70DB.98E1.3DEC#copy pcap

AP70DB.98E1.3DEC_capture.pcap0

 tftp 192.168.1.100
#########################################################################################################
AP70DB.98E1.3DEC#
```

Step 4. Now you can open the file in wireshark. The file is pcap0. Change to pcap so that it automatically associates with wireshark.

Command Options

The debug traffic wired command has several options that can help you to capture specific traffic:

```
APC4F7.D54C.E77C#debug traffic wired
  <0-3>   wired debug interface number
  filter  filter packets with tcpdump  filter string
  ip      Enable wired ip traffic dump
  tcp     Enable wired tcp traffic dump
  udp     Enable wired udp traffic dum
```

You can add "verbose" at the end of the debug command to see the hex dump of the packet. Be aware that this can overwhelm your CLI session very quickly if your filter is not narrow enough.

Wired PCAP through the use of Filter

The filter format corresponds with tcpdump capture filter format.

	Filter Example	Description
Host	“host 192.168.2.5”	This filters the packet capture to only gather packets which go to or come from the host 192.168.2.5.

	“src host 192.168.2.5”	This filters the packet capture to only gather packets that come from 192.168.2.5.
	“dst host 192.168.2.5“	This filters the packet capture to only gather packets which go to 192.168.2.5.
Port	“port 443”	This filters the packet capture to only gather packets with a source or destination of port 443.
	“src port 1055”	This captures traffic which is sourced from port 1055.
	“dst port 443”	This captures traffic destined for port 443.

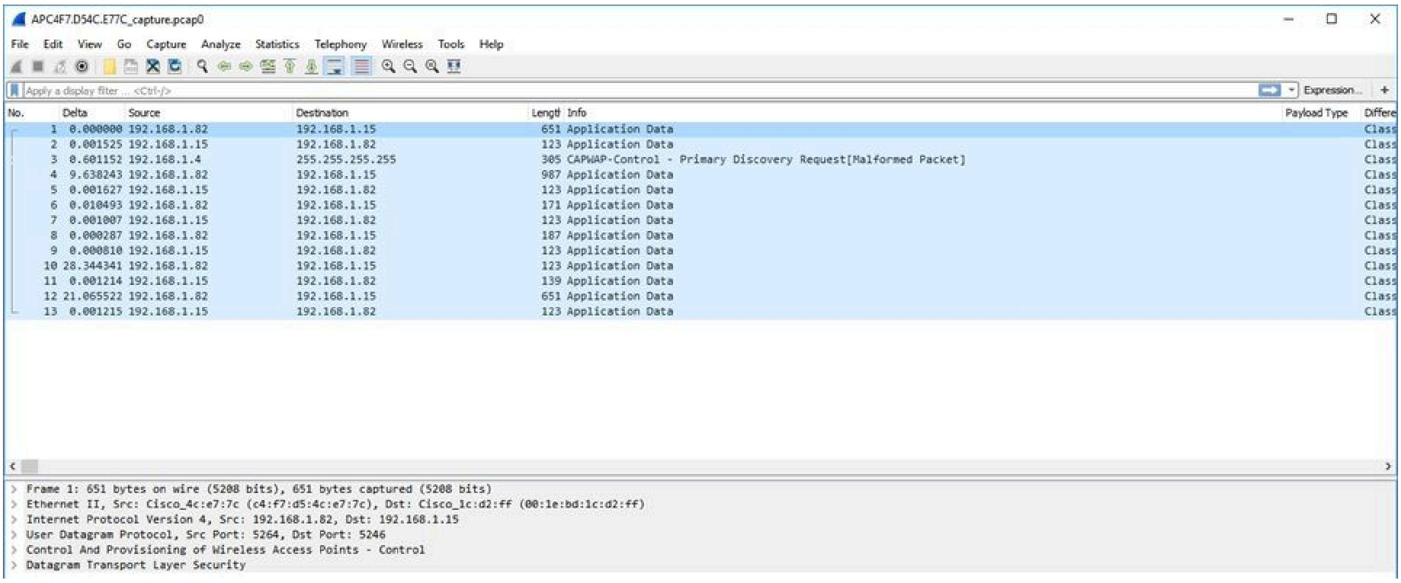
Here is an example where the output displays on the console but also filtered to only see CAPWAP data packets:

```
APC4F7.D54C.E77C#debug traffic wired filter "port 5246"
APC4F7.D54C.E77C#reading from file /dev/click_wired_log, link-type EN10MB (Ethernet)
12:20:50.483125 IP APC4F7-D54C-E77C.lan.5264 > 192.168.1.15.5246: UDP, length 81
12:20:50.484361 IP 192.168.1.15.5246 > APC4F7-D54C-E77C.lan.5264: UDP, length 97

APC4F7.D54C.E77C#no debug traffic wired filter "port 5246"
APC4F7.D54C.E77C#Killed
APC4F7.D54C.E77C#
```

Example of output on File:

```
APC4F7.D54C.E77C#debug traffic wired filter "port 5246" capture
% Writing packets to "/tmp/pcap/APC4F7.D54C.E77C_capture.pcap0"
APC4F7.D54C.E77C#reading from file /dev/click_wired_log, link-type EN10MB (Ethernet)
APC4F7.D54C.E77C#no debug traffic wired filter "port 5246" capture
APC4F7.D54C.E77C#copy pcap APC4F7.D54C.E77C_capture.pcap0 tftp 192.168.1.100
#######################################################################################################
APC4F7.D54C.E77C#
```

To open the capture on wireshark:



APC4F7.D54C.E77C_capture.pcap0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Expression…
+

No.	Delta	Source	Destination	Lengtl	Info	Payload Type	Differe
1	0.000000	192.168.1.82	192.168.1.15	651	Application Data		Class
2	0.001525	192.168.1.15	192.168.1.82	123	Application Data		Class
3	0.601152	192.168.1.4	255.255.255.255	305	CAPWAP-Control - Primary Discovery Request[Malformed Packet]		Class
4	9.638243	192.168.1.82	192.168.1.15	987	Application Data		Class
5	0.001627	192.168.1.15	192.168.1.82	123	Application Data		Class
6	0.010493	192.168.1.82	192.168.1.15	171	Application Data		Class
7	0.001007	192.168.1.15	192.168.1.82	123	Application Data		Class
8	0.000287	192.168.1.82	192.168.1.15	187	Application Data		Class
9	0.000810	192.168.1.15	192.168.1.82	123	Application Data		Class
10	28.344341	192.168.1.82	192.168.1.15	123	Application Data		Class
11	0.001214	192.168.1.15	192.168.1.82	139	Application Data		Class
12	21.065522	192.168.1.82	192.168.1.15	651	Application Data		Class
13	0.001215	192.168.1.15	192.168.1.82	123	Application Data		Class

```
> Frame 1: 651 bytes on wire (5208 bits), 651 bytes captured (5208 bits)
> Ethernet II, Src: Cisco_4c:e7:7c (c4:f7:d5:4c:e7:7c), Dst: Cisco_1c:d2:ff (00:1e:bd:1c:d2:ff)
> Internet Protocol Version 4, Src: 192.168.1.82, Dst: 192.168.1.15
> User Datagram Protocol, Src Port: 5264, Dst Port: 5246
> Control And Provisioning of Wireless Access Points - Control
> Datagram Transport Layer Security
```

Radio Capture

It is possible to enable the capture of packets on the control-plane of the radio. Due to performance impact, it is not possible to capture on the radio dataplane.

This means that the client association flow (probes, authentication, association, eap, arp, dhcp packets as well as ipv6 control packets, icmp and ndp) is visible but not the data the client passes after the move to the connected state.

Procedure

Step 1. Add the tracked client mac address. Several mac addresses can be added. It is also possible to run the command for all clients but this is not recommended.

```
config ap client-trace address add < client-mac> --- Per client debugging. Allows multiple macs.
config ap client-trace all-clients <enable | disable>  -- All clients debugging. Not recommended.
```

Step 2. Set a filter to only log specific protocols or all supported protocols:

```
config ap client-trace filter <all|arp|assoc|auth|dhcp|eap|icmp|ipv6|ndp|probe> <enable|disable>
```

Step 3. Chose to display the output on console (asynchronously):

```
configure ap client-trace output console-log enable
```

Step 4. Start the trace.

```
config ap client-trace start
```

Example:

<#root>

```
AP0CD0.F894.46E4#show dot11 clients

Total dot11 clients: 1
       Client MAC Slot ID WLAN ID AID    WLAN Name RSSI Maxrate WGB

A8:DB:03:08:4C:4A

       0       1   1 testewlcwlan  -41 MCS92SS  No

AP0CD0.F894.46E4#config ap client-trace address add

A8:DB:03:08:4C:4A


AP0CD0.F894.46E4#config ap client-trace filter
  all    Trace ALL filters
  arp    Trace arp Packets
  assoc  Trace assoc Packets
  auth   Trace auth Packets
  dhcp   Trace dhcp Packets
  eap    Trace eap Packets
  icmp   Trace icmp Packets
  ipv6   Trace IPv6 Packets
  ndp    Trace ndp Packets
  probe  Trace probe Packets
AP0CD0.F894.46E4#config ap client-trace filter all enable
AP0CD0.F894.46E4#configure ap client-trace output console-log enable
AP0CD0.F894.46E4#configure ap client-trace start
AP0CD0.F894.46E4#term mon
```

To stop the capture:

```
configure ap client-trace stop
configure ap client-trace clear
configure ap client-trace address clear
```

Verify

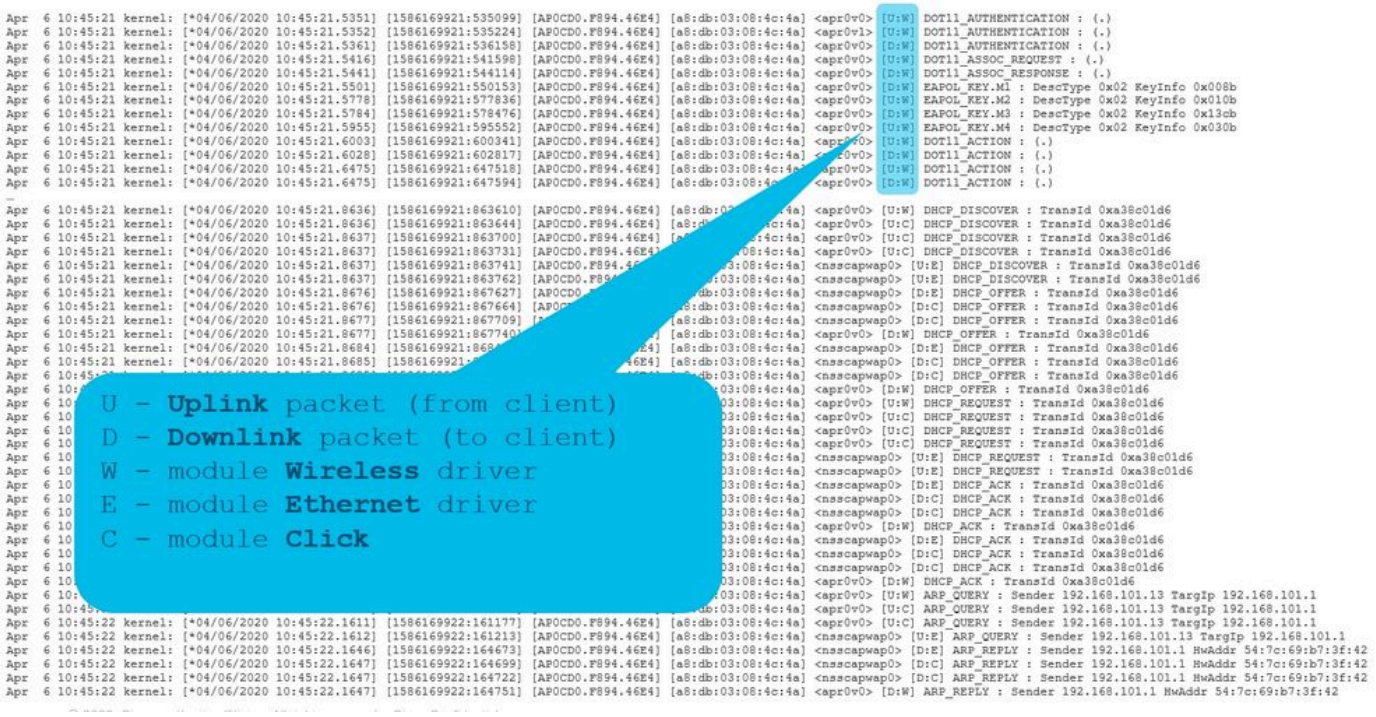
Verify Client Trace:

<#root>

```
AP70DB.98E1.3DEC#

show ap client-trace status
```

```
Client Trace Status         : Started

Client Trace ALL Clients    : disable

Client Trace Address        : a8:db:03:08:4c:4a

Remote/Dump Client Trace Address    : a8:db:03:08:4c:4a

Client Trace Filter         : probe
Client Trace Filter         : auth
Client Trace Filter         : assoc
Client Trace Filter         : eap
Client Trace Filter         : dhcp
Client Trace Filter         : dhcpv6
Client Trace Filter         : icmp
Client Trace Filter         : icmpv6
Client Trace Filter         : ndp
Client Trace Filter         : arp

Client Trace Output         : eventbuf
Client Trace Output         : console-log
Client Trace Output         : dump
Client Trace Output         : remote

Remote trace IP             : 192.168.1.100
Remote trace dest port      : 5688
NOTE - Only VIP packets are seen on remote if VIP is enabled

Dump packet length          : 10
Client Trace Inline Monitor           : disable
Client Trace Inline Monitor pkt-attach : disable
```

Example of a successful client connection:



```
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.5351] [1586169921:535099] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [U:W] DOT11_AUTHENTICATION : (.)
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.5352] [1586169921:535224] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v1> [U:W] DOT11_AUTHENTICATION : (.)
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.5361] [1586169921:536158] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [D:W] DOT11_AUTHENTICATION : (.)
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.5416] [1586169921:541598] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [U:W] DOT11_ASSOC_REQUEST : (.)
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.5441] [1586169921:544114] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [D:W] DOT11_ASSOC_RESPONSE : (.)
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.5501] [1586169921:550153] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [D:W] EAPOL_KEY.M1 : DescType 0x02 KeyInfo 0x008b
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.5778] [1586169921:577836] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [U:W] EAPOL_KEY.M2 : DescType 0x02 KeyInfo 0x010b
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.5784] [1586169921:578476] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [D:W] EAPOL_KEY.M3 : DescType 0x02 KeyInfo 0x13cb
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.5955] [1586169921:595552] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [U:W] EAPOL_KEY.M4 : DescType 0x02 KeyInfo 0x030b
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.6003] [1586169921:600341] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr v0> [U:W] DOT11_ACTION : (.)
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.6028] [1586169921:602817] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <  r0v0> [D:W] DOT11_ACTION : (.)
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.6475] [1586169921:647518] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a  apr0v0> [U:W] DOT11_ACTION : (.)
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.6475] [1586169921:647594] [AP0CD0.F894.46E4] [a8:db:03:08:4c      <apr0v0> [D:W] DOT11_ACTION : (.)
—
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.8636] [1586169921:863610] [AP0CD0.F894.46E4] [a8:db:0     4a] <apr0v0> [U:W] DHCP_DISCOVER : TransId 0xa38c01d6
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.8636] [1586169921:863644] [AP0CD0.F894.46E4] [a8:d     c:4a] <apr0v0> [U:C] DHCP_DISCOVER : TransId 0xa38c01d6
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.8637] [1586169921:863700] [AP0CD0.F894.46E4] [a     :4c:4a] <apr0v0> [U:C] DHCP_DISCOVER : TransId 0xa38c01d6
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.8637] [1586169921:863731] [AP0CD0.F894.46E4     08:4c:4a] <apr0v0> [U:C] DHCP_DISCOVER : TransId 0xa38c01d6
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.8637] [1586169921:863741] [AP0CD0.F894.4     3:08:4c:4a] <nsscapwap0> [U:E] DHCP_DISCOVER : TransId 0xa38c01d6
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.8637] [1586169921:863762] [AP0CD0.F89     :03:08:4c:4a] <nsscapwap0> [U:E] DHCP_DISCOVER : TransId 0xa38c01d6
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.8676] [1586169921:867627] [AP0CD0.     db:03:08:4c:4a] <nsscapwap0> [D:E] DHCP_OFFER : TransId 0xa38c01d6
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.8676] [1586169921:867664] [AP0C     8:db:03:08:4c:4a] <nsscapwap0> [D:C] DHCP_OFFER : TransId 0xa38c01d6
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.8677] [1586169921:867709]      [a8:db:03:08:4c:4a] <nsscapwap0> [D:C] DHCP_OFFER : TransId 0xa38c01d6
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.8677] [1586169921:867740     [a8:db:03:08:4c:4a] <apr0v0> [D:W] DHCP_OFFER : TransId 0xa38c01d6
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.8684] [1586169921:868     E4] [a8:db:03:08:4c:4a] <nsscapwap0> [D:E] DHCP_OFFER : TransId 0xa38c01d6
Apr  6 10:45:21 kernel: [*04/06/2020 10:45:21.8685] [1586169921:     6E4] [a8:db:03:08:4c:4a] <nsscapwap0> [D:C] DHCP_OFFER : TransId 0xa38c01d6
Apr  6 10:45     db:03:08:4c:4a] <nsscapwap0> [D:C] DHCP_OFFER : TransId 0xa38c01d6
Apr  6 10:     03:08:4c:4a] <apr0v0> [D:W] DHCP_OFFER : TransId 0xa38c01d6
Apr  6 10     03:08:4c:4a] <apr0v0> [U:W] DHCP_REQUEST : TransId 0xa38c01d6
Apr  6 10     03:08:4c:4a] <apr0v0> [U:C] DHCP_REQUEST : TransId 0xa38c01d6
Apr  6 10     03:08:4c:4a] <apr0v0> [U:C] DHCP_REQUEST : TransId 0xa38c01d6
Apr  6 10     03:08:4c:4a] <apr0v0> [U:C] DHCP_REQUEST : TransId 0xa38c01d6
Apr  6 10     03:08:4c:4a] <nsscapwap0> [U:E] DHCP_REQUEST : TransId 0xa38c01d6
Apr  6 10     03:08:4c:4a] <nsscapwap0> [U:E] DHCP_REQUEST : TransId 0xa38c01d6
Apr  6 10     03:08:4c:4a] <nsscapwap0> [D:E] DHCP_ACK : TransId 0xa38c01d6
Apr  6 10     03:08:4c:4a] <nsscapwap0> [D:C] DHCP_ACK : TransId 0xa38c01d6
Apr  6 10     03:08:4c:4a] <nsscapwap0> [D:C] DHCP_ACK : TransId 0xa38c01d6
Apr  6 10     03:08:4c:4a] <apr0v0> [D:W] DHCP_ACK : TransId 0xa38c01d6
Apr  6 10     03:08:4c:4a] <nsscapwap0> [D:E] DHCP_ACK : TransId 0xa38c01d6
Apr  6 10     03:08:4c:4a] <nsscapwap0> [D:C] DHCP_ACK : TransId 0xa38c01d6
Apr  6 10     03:08:4c:4a] <nsscapwap0> [D:C] DHCP_ACK : TransId 0xa38c01d6
Apr  6 10     03:08:4c:4a] <apr0v0> [D:W] DHCP_ACK : TransId 0xa38c01d6
Apr  6 10:     :03:08:4c:4a] <apr0v0> [U:W] ARP_QUERY : Sender 192.168.101.13 TargIp 192.168.101.1
Apr  6 10:45     db:03:08:4c:4a] <apr0v0> [U:C] ARP_QUERY : Sender 192.168.101.13 TargIp 192.168.101.1
Apr  6 10:45:22 kernel: [*04/06/2020 10:45:22.1611] [1586169922:161177] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [U:C] ARP_QUERY : Sender 192.168.101.13 TargIp 192.168.101.1
Apr  6 10:45:22 kernel: [*04/06/2020 10:45:22.1612] [1586169922:161213] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <nsscapwap0> [U:E] ARP_QUERY : Sender 192.168.101.13 TargIp 192.168.101.1
Apr  6 10:45:22 kernel: [*04/06/2020 10:45:22.1646] [1586169922:164673] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <nsscapwap0> [D:E] ARP_REPLY : Sender 192.168.101.1 HwAddr 54:7c:69:b7:3f:42
Apr  6 10:45:22 kernel: [*04/06/2020 10:45:22.1647] [1586169922:164699] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <nsscapwap0> [D:C] ARP_REPLY : Sender 192.168.101.1 HwAddr 54:7c:69:b7:3f:42
Apr  6 10:45:22 kernel: [*04/06/2020 10:45:22.1647] [1586169922:164722] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <nsscapwap0> [D:C] ARP_REPLY : Sender 192.168.101.1 HwAddr 54:7c:69:b7:3f:42
Apr  6 10:45:22 kernel: [*04/06/2020 10:45:22.1647] [1586169922:164751] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [D:W] ARP_REPLY : Sender 192.168.101.1 HwAddr 54:7c:69:b7:3f:42
```

U - **Uplink** packet (from client)
D - **Downlink** packet (to client)
W - module **Wireless** driver
E - module **Ethernet** driver
C - module **Click**

The letters between brackets help you understand where that frame was seen (E for Ethernet, W for Wireless, C for the Click module when it is internal to the AP) and in which direction (Upload or

Download).

Here is a small table of the meaning of those letters:

U - uplink packet(from client)
D - downlink packet(to click)
W - module wireless driver
E - module Ethernet driver
C - module Click

Other Options

View Log asyncronously:

The logs can then be consulted with the command: "**show ap client-trace events mac xx:xx:xx:xx:xx:xx"**
(or replace the mac with "all")

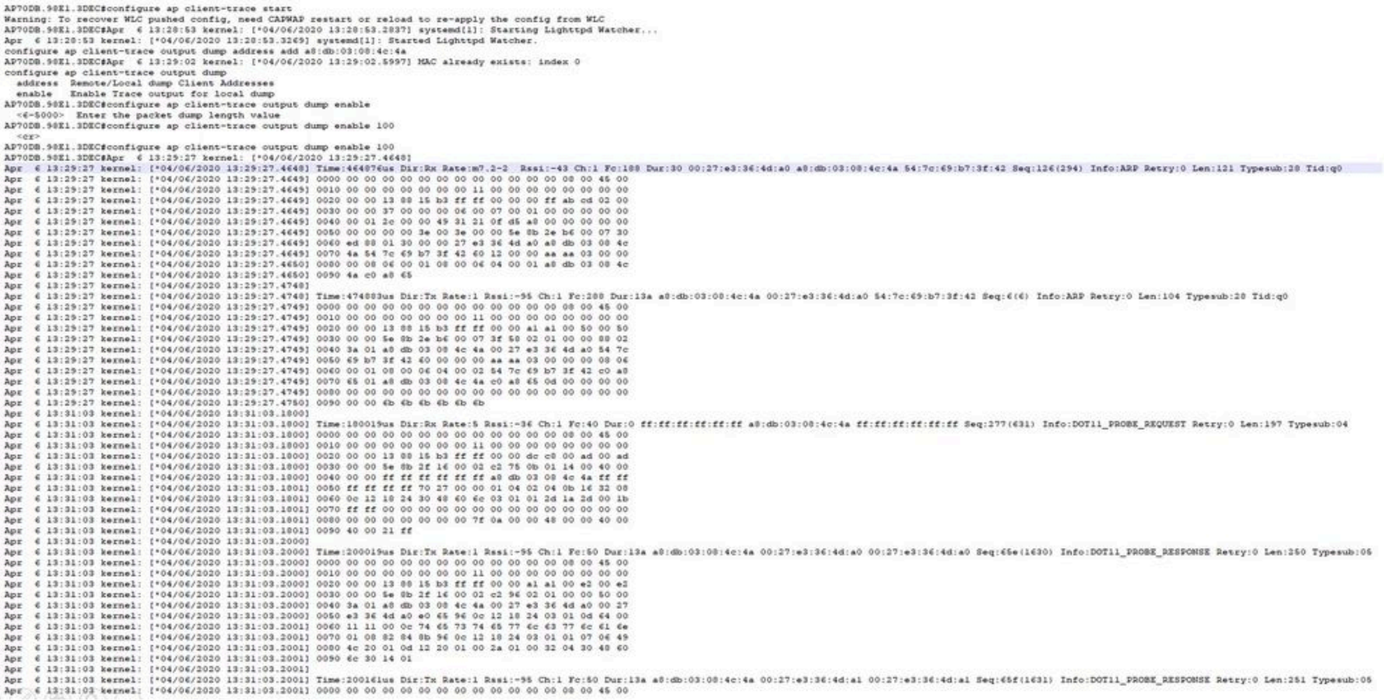
<#root>

```
AP0CD0.F894.46E4#

show ap client-trace events mac a8:db:03:08:4c:4a


[*04/06/2020 10:11:54.287675] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v1> [U:W] DOT11_AUTHENTICATIO
[*04/06/2020 10:11:54.288144] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [D:W] DOT11_AUTHENTICATIO
[*04/06/2020 10:11:54.289870] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [U:W] DOT11_ASSOC_REQUEST
[*04/06/2020 10:11:54.317341] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [D:W] DOT11_ASSOC_RESPONS
[*04/06/2020 10:11:54.341370] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [D:W] EAPOL_KEY.M1 : Desc
[*04/06/2020 10:11:54.374500] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [U:W] EAPOL_KEY.M2 : Desc
[*04/06/2020 10:11:54.377237] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [D:W] EAPOL_KEY.M3 : Desc
[*04/06/2020 10:11:54.390255] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [U:W] EAPOL_KEY.M4 : Desc
[*04/06/2020 10:11:54.396855] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [U:W] DOT11_ACTION : (.)
[*04/06/2020 10:11:54.416650] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [D:W] DOT11_ACTION : (.)
[*04/06/2020 10:11:54.469089] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [U:W] DOT11_ACTION : (.)
[*04/06/2020 10:11:54.469157] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [D:W] DOT11_ACTION : (.)
[*04/06/2020 10:11:57.921877] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [U:W] DOT11_ACTION : (.)
[*04/06/2020 10:11:57.921942] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [D:W] DOT11_ACTION : (.)
[*04/06/2020 10:15:36.123119] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [D:W] DOT11_DEAUTHENTICAT
[*04/06/2020 10:15:36.127731] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr1v0> [D:W] DOT11_DISASSOC : (.
[*04/06/2020 10:17:24.128751] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [U:W] DOT11_AUTHENTICATIO
[*04/06/2020 10:17:24.128870] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v1> [U:W] DOT11_AUTHENTICATIO
[*04/06/2020 10:17:24.129303] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [D:W] DOT11_AUTHENTICATIO
[*04/06/2020 10:17:24.133026] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [U:W] DOT11_ASSOC_REQUEST
[*04/06/2020 10:17:24.136095] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [D:W] DOT11_ASSOC_RESPONS
[*04/06/2020 10:17:24.138732] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [D:W] EAPOL_KEY.M1 : Desc
[*04/06/2020 10:17:24.257295] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [U:W] EAPOL_KEY.M2 : Desc
[*04/06/2020 10:17:24.258105] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [D:W] EAPOL_KEY.M3 : Desc
[*04/06/2020 10:17:24.278937] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [U:W] EAPOL_KEY.M4 : Desc
[*04/06/2020 10:17:24.287459] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [U:W] DOT11_ACTION : (.)
[*04/06/2020 10:17:24.301344] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [D:W] DOT11_ACTION : (.)
[*04/06/2020 10:17:24.327482] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [U:W] DOT11_ACTION : (.)
[*04/06/2020 10:17:24.327517] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [D:W] DOT11_ACTION : (.)
[*04/06/2020 10:17:24.430136] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [U:W] DOT11_ACTION : (.)
[*04/06/2020 10:17:24.430202] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [D:W] DOT11_ACTION : (.)
[*04/06/2020 10:19:08.075326] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [U:W] DOT11_PROBE_REQUEST
[*04/06/2020 10:19:08.075392] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v0> [D:W] DOT11_PROBE_RESPONS
[*04/06/2020 10:19:08.075437] [AP0CD0.F894.46E4] [a8:db:03:08:4c:4a] <apr0v1> [U:W] DOT11_PROBE_REQUEST
```

Dump the packets in hex format

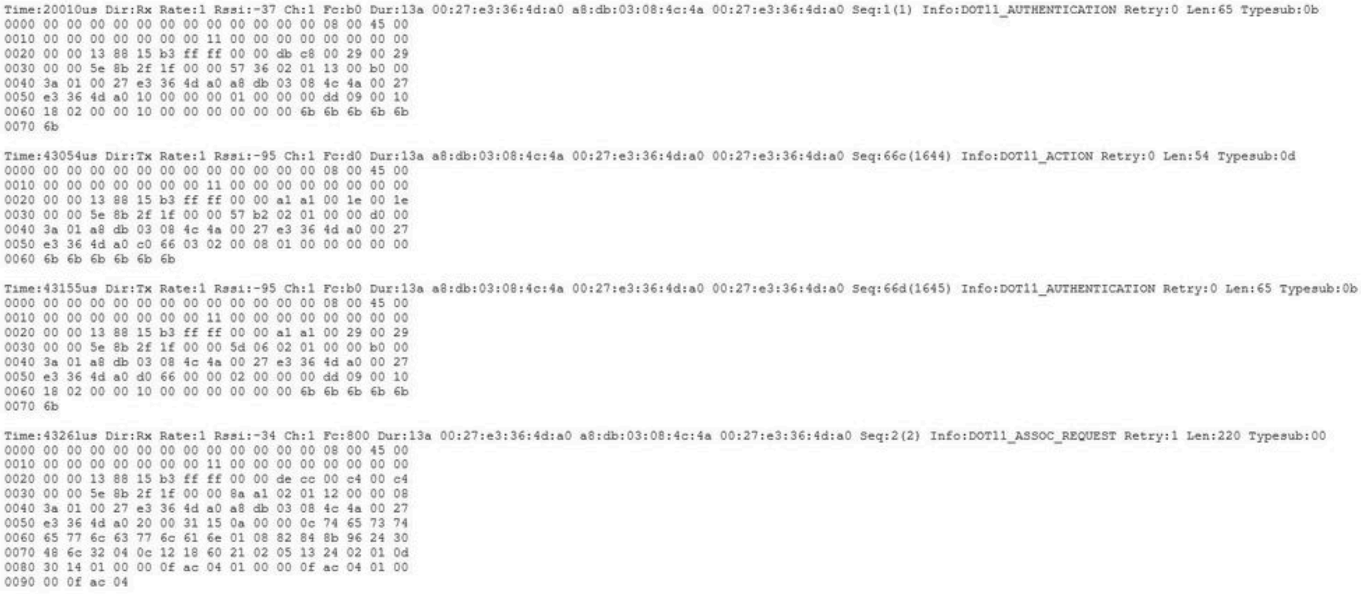
You can dump the packets in hex format in the CLI:

```
configure ap client-trace output dump address add xx:xx:xx:xx:xx:xx
configure ap client-trace output dump enable x -> Enter the packet dump length value
```

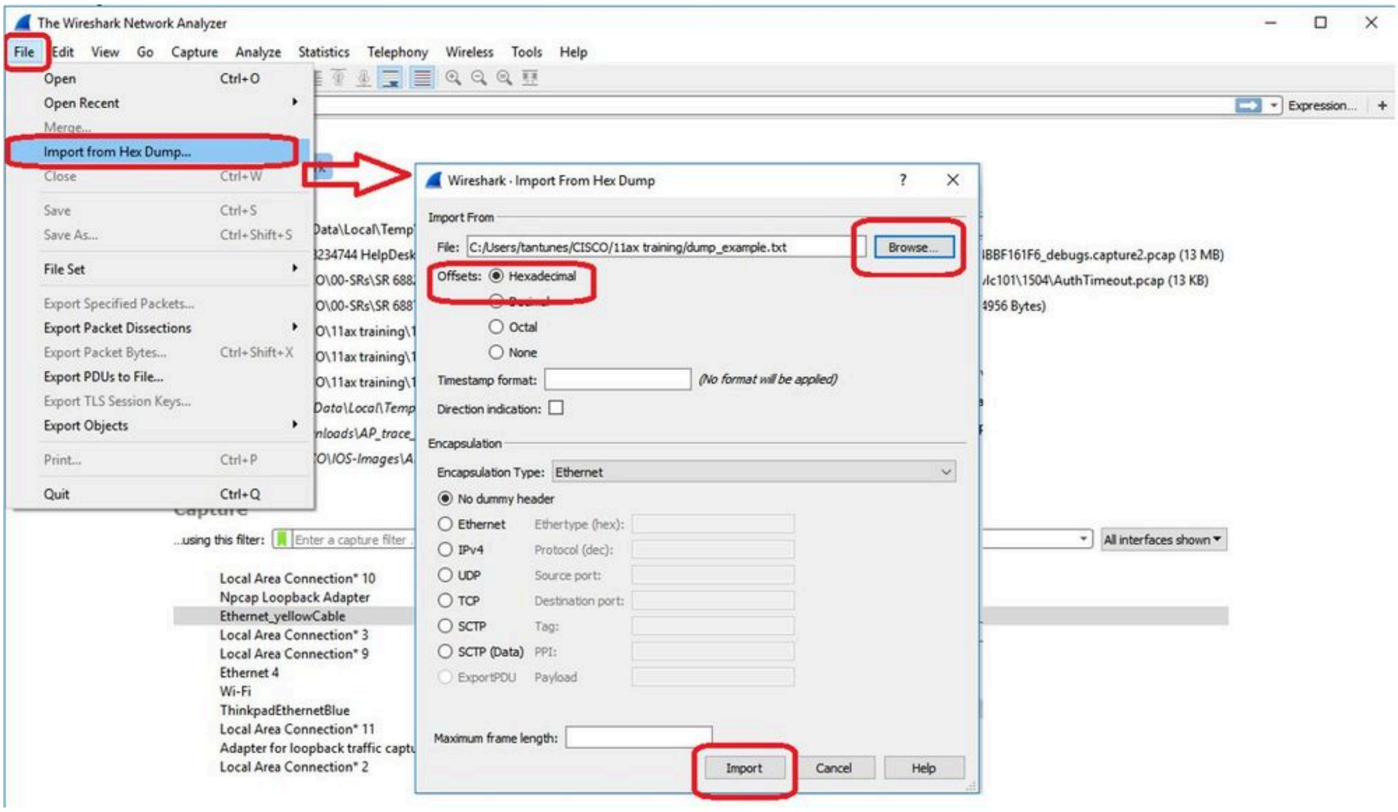


```
AP70DB.98E1.3DEC#configure ap client-trace start
Warning: To recover WLC pushed config, need CAPWAP restart or reload to re-apply the config from WLC
AP70DB.98E1.3DEC#Apr  6 13:28:53 kernel: [*04/06/2020 13:28:53.2837] systemd[1]: Starting Lighttpd Watcher...
Apr  6 13:28:53 kernel: [*04/06/2020 13:28:53.3269] systemd[1]: Started Lighttpd Watcher.
configure ap client-trace output dump address add a8:db:03:08:4c:4a
AP70DB.98E1.3DEC#Apr  6 13:29:02 kernel: [*04/06/2020 13:29:02.5997] MAC already exists: index 0
configure ap client-trace output dump
  address  Remote/Local dump Client Addresses
  enable   Enable Trace output for local dump
AP70DB.98E1.3DEC#configure ap client-trace output dump enable
  <4-5000>  Enter the packet dump length value
AP70DB.98E1.3DEC#configure ap client-trace output dump enable 100
  <cr>
AP70DB.98E1.3DEC#configure ap client-trace output dump enable 100
AP70DB.98E1.3DEC#Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4648]
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4648] Time:464076us Dir:Rx Rate:m7.2-2  Rssi:-43 Ch:1 Fc:188 Dur:30 00:27:e3:36:4d:a0 a8:db:03:08:4c:4a 54:7c:69:b7:3f:42 Seq:126(294) Info:ARP Retry:0 Len:121 Typesub:28 Tid:q0
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4649] 0000 00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4649] 0010 00 00 00 00 00 00 00 11 00 00 00 00 00 00 00 00
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4649] 0020 00 00 13 88 15 b3 ff ff 00 00 00 ff ab cd 02 00
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4649] 0030 00 00 37 00 00 00 06 00 07 00 01 00 00 00 00 00
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4649] 0040 00 01 2c 00 00 49 31 21 0f d5 a8 00 00 00 00 00
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4649] 0050 00 00 00 00 3e 00 3e 00 00 5e 8b 2e b6 00 07 30
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4649] 0060 ed 88 01 30 00 00 27 e3 36 4d a0 a8 db 03 08 4c
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4649] 0070 4a 54 7c 69 b7 3f 42 60 12 00 00 aa aa 03 00 00
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4650] 0080 00 08 06 00 01 08 00 06 04 00 01 a8 db 03 08 4c
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4650] 0090 4a c0 a8 65
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4748]
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4748] Time:474883us Dir:Tx Rate:1 Rssi:-95 Ch:1 Fc:288 Dur:13a a8:db:03:08:4c:4a 00:27:e3:36:4d:a0 54:7c:69:b7:3f:42 Seq:6(6) Info:ARP Retry:0 Len:104 Typesub:28 Tid:q0
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4749] 0000 00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4749] 0010 00 00 00 00 00 00 00 11 00 00 00 00 00 00 00 00
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4749] 0020 00 00 13 88 15 b3 ff ff 00 00 a1 a1 00 50 00 50
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4749] 0030 00 00 5e 8b 2e b6 00 07 3f 58 02 01 00 00 88 02
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4749] 0040 3a 01 a8 db 03 08 4c 4a 00 27 e3 36 4d a0 54 7c
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4749] 0050 69 b7 3f 42 60 00 00 00 aa aa 03 00 00 00 08 06
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4749] 0060 00 01 08 00 06 04 00 02 54 7c 69 b7 3f 42 c0 a8
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4749] 0070 65 01 a8 db 03 08 4c 4a c0 a8 65 0d 00 00 00 00
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4749] 0080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr  6 13:29:27 kernel: [*04/06/2020 13:29:27.4750] 0090 00 00 6b 6b 6b 6b 6b 6b
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.1800]
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.1800] Time:180019us Dir:Rx Rate:5 Rssi:-36 Ch:1 Fc:40 Dur:0 ff:ff:ff:ff:ff:ff a8:db:03:08:4c:4a ff:ff:ff:ff:ff:ff Seq:277(631) Info:DOT11_PROBE_REQUEST Retry:0 Len:197 Typesub:04
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.1800] 0000 00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.1800] 0010 00 00 00 00 00 00 00 11 00 00 00 00 00 00 00 00
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.1800] 0020 00 00 13 88 15 b3 ff ff 00 00 dc c8 00 ad 00 ad
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.1800] 0030 00 00 5e 8b 2f 16 00 02 c2 75 0b 01 14 00 40 00
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.1800] 0040 00 00 ff ff ff ff ff ff a8 db 03 08 4c 4a ff ff
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.1801] 0050 ff ff ff ff 70 27 00 00 01 04 02 04 0b 16 32 08
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.1801] 0060 0c 12 18 24 30 48 60 6c 03 01 01 2d 1a 2d 00 1b
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.1801] 0070 ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.1801] 0080 00 00 00 00 00 00 00 7f 0a 00 00 48 00 00 40 00
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.1801] 0090 40 00 21 ff
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.2000]
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.2000] Time:200019us Dir:Tx Rate:1 Rssi:-95 Ch:1 Fc:50 Dur:13a a8:db:03:08:4c:4a 00:27:e3:36:4d:a0 00:27:e3:36:4d:a0 Seq:65e(1630) Info:DOT11_PROBE_RESPONSE Retry:0 Len:250 Typesub:05
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.2000] 0000 00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.2000] 0010 00 00 00 00 00 00 00 11 00 00 00 00 00 00 00 00
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.2000] 0020 00 00 13 88 15 b3 ff ff 00 00 a1 a1 00 e2 00 e2
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.2000] 0030 00 00 5e 8b 2f 16 00 02 c2 96 02 01 00 00 50 00
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.2000] 0040 3a 01 a8 db 03 08 4c 4a 00 27 e3 36 4d a0 00 27
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.2000] 0050 e3 36 4d a0 e0 65 96 0c 12 18 24 03 01 0d 64 00
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.2001] 0060 11 11 00 0c 74 65 73 74 65 77 6c 63 77 6c 61 6e
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.2001] 0070 01 08 82 84 8b 96 0c 12 18 24 03 01 01 07 06 49
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.2001] 0080 4c 20 01 0d 12 20 01 00 2a 01 00 32 04 30 48 60
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.2001] 0090 6c 30 14 01
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.2001]
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.2001] Time:200161us Dir:Tx Rate:1 Rssi:-95 Ch:1 Fc:50 Dur:13a a8:db:03:08:4c:4a 00:27:e3:36:4d:a1 00:27:e3:36:4d:a1 Seq:65f(1631) Info:DOT11_PROBE_RESPONSE Retry:0 Len:251 Typesub:05
Apr  6 13:31:03 kernel: [*04/06/2020 13:31:03.2001] 0000 00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00
```

Then, you can clean the hex dump and save in txt format and import into wireshark:



```
Time:20010us Dir:Rx Rate:1 Rssi:-37 Ch:1 Fc:b0 Dur:13a 00:27:e3:36:4d:a0 a8:db:03:08:4c:4a 00:27:e3:36:4d:a0 Seq:1(1) Info:DOT11_AUTHENTICATION Retry:0 Len:65 Typesub:0b
0000 00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00
0010 00 00 00 00 00 00 00 11 00 00 00 00 00 00 00 00
0020 00 00 13 88 15 b3 ff ff 00 00 db c8 00 29 00 29
0030 00 00 5e 8b 2f 1f 00 00 57 36 02 01 13 00 b0 00
0040 3a 01 00 27 e3 36 4d a0 a8 db 03 08 4c 4a 00 27
0050 e3 36 4d a0 10 00 00 00 01 00 00 00 dd 09 00 10
0060 18 02 00 00 10 00 00 00 00 00 00 6b 6b 6b 6b 6b
0070 6b

Time:43054us Dir:Tx Rate:1 Rssi:-95 Ch:1 Fc:d0 Dur:13a a8:db:03:08:4c:4a 00:27:e3:36:4d:a0 00:27:e3:36:4d:a0 Seq:66c(1644) Info:DOT11_ACTION Retry:0 Len:54 Typesub:0d
0000 00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00
0010 00 00 00 00 00 00 00 11 00 00 00 00 00 00 00 00
0020 00 00 13 88 15 b3 ff ff 00 00 a1 a1 00 1e 00 1e
0030 00 00 5e 8b 2f 1f 00 00 57 b2 02 01 00 00 d0 00
0040 3a 01 a8 db 03 08 4c 4a 00 27 e3 36 4d a0 00 27
0050 e3 36 4d a0 c0 66 03 02 00 08 01 00 00 00 00 00
0060 6b 6b 6b 6b 6b 6b

Time:43155us Dir:Tx Rate:1 Rssi:-95 Ch:1 Fc:b0 Dur:13a a8:db:03:08:4c:4a 00:27:e3:36:4d:a0 00:27:e3:36:4d:a0 Seq:66d(1645) Info:DOT11_AUTHENTICATION Retry:0 Len:65 Typesub:0b
0000 00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00
0010 00 00 00 00 00 00 00 11 00 00 00 00 00 00 00 00
0020 00 00 13 88 15 b3 ff ff 00 00 a1 a1 00 29 00 29
0030 00 00 5e 8b 2f 1f 00 00 5d 06 02 01 00 00 b0 00
0040 3a 01 a8 db 03 08 4c 4a 00 27 e3 36 4d a0 00 27
0050 e3 36 4d a0 d0 66 00 00 02 00 00 00 dd 09 00 10
0060 18 02 00 00 10 00 00 00 00 00 00 6b 6b 6b 6b 6b
0070 6b

Time:43261us Dir:Rx Rate:1 Rssi:-34 Ch:1 Fc:800 Dur:13a 00:27:e3:36:4d:a0 a8:db:03:08:4c:4a 00:27:e3:36:4d:a0 Seq:2(2) Info:DOT11_ASSOC_REQUEST Retry:1 Len:220 Typesub:00
0000 00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00
0010 00 00 00 00 00 00 00 11 00 00 00 00 00 00 00 00
0020 00 00 13 88 15 b3 ff ff 00 00 de cc 00 c4 00 c4
0030 00 00 5e 8b 2f 1f 00 00 8a a1 02 01 12 00 00 08
0040 3a 01 00 27 e3 36 4d a0 a8 db 03 08 4c 4a 00 27
0050 e3 36 4d a0 20 00 31 15 0a 00 00 0c 74 65 73 74
0060 65 77 6c 63 77 6c 61 6e 01 08 82 84 8b 96 24 30
0070 48 6c 32 04 0c 12 18 60 21 02 05 13 24 02 01 0d
0080 30 14 01 00 00 0f ac 04 01 00 00 0f ac 04 01 00
0090 00 0f ac 04
```



File: C:/Users/tantunes/CISCO/11ax training/dump_example.txt

Offsets: ● Hexadecimal ○ Decimal ○ Octal ○ None

Timestamp format: (No format will be applied)

Direction indication: □

Encapsulation Type: Ethernet

● No dummy header
○ Ethernet Ethertype (hex):
○ IPv4 Protocol (dec):
○ UDP Source port:
○ TCP Destination port:
○ SCTP Tag:
○ SCTP (Data) PPI:
○ ExportPDU Payload

Maximum frame length:

Because the output can be very large and to consider that the output only mentions what frame type is seen and not any of the inner detail, it can be more efficient to redirect the packet capture to a laptop that run a a capture application (such as wireshark).

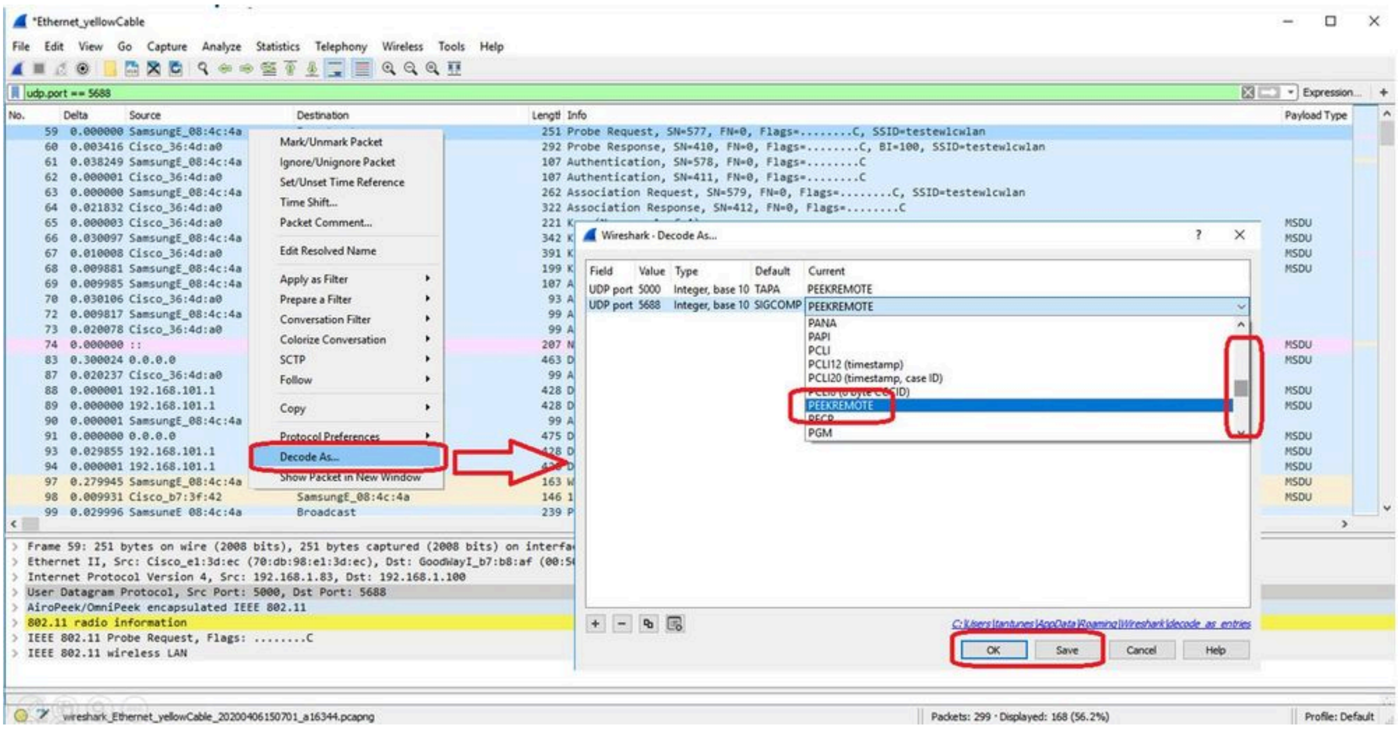
Enable the remote capture feature to send the packets to external device with wireshark:

```
config ap client-trace output remote enable
```

The command means the AP forwards every frame captured by the client-trace filter towards the laptop at 192.168.68.68 and uses PEEKREMOTE encapsulation (just like APs in sniffer mode) on port 5000.

One limitation is that the target laptop has to be in the same subnet as the AP where you run this command on. You can change the port number to accomodate any security policies in place in your network.

Once you received all the packets on the laptop that runs Wireshark, you can right click on the udp 5000 header and chose **decode as** and pick PEEKREMOTE as illustrated in this figure:

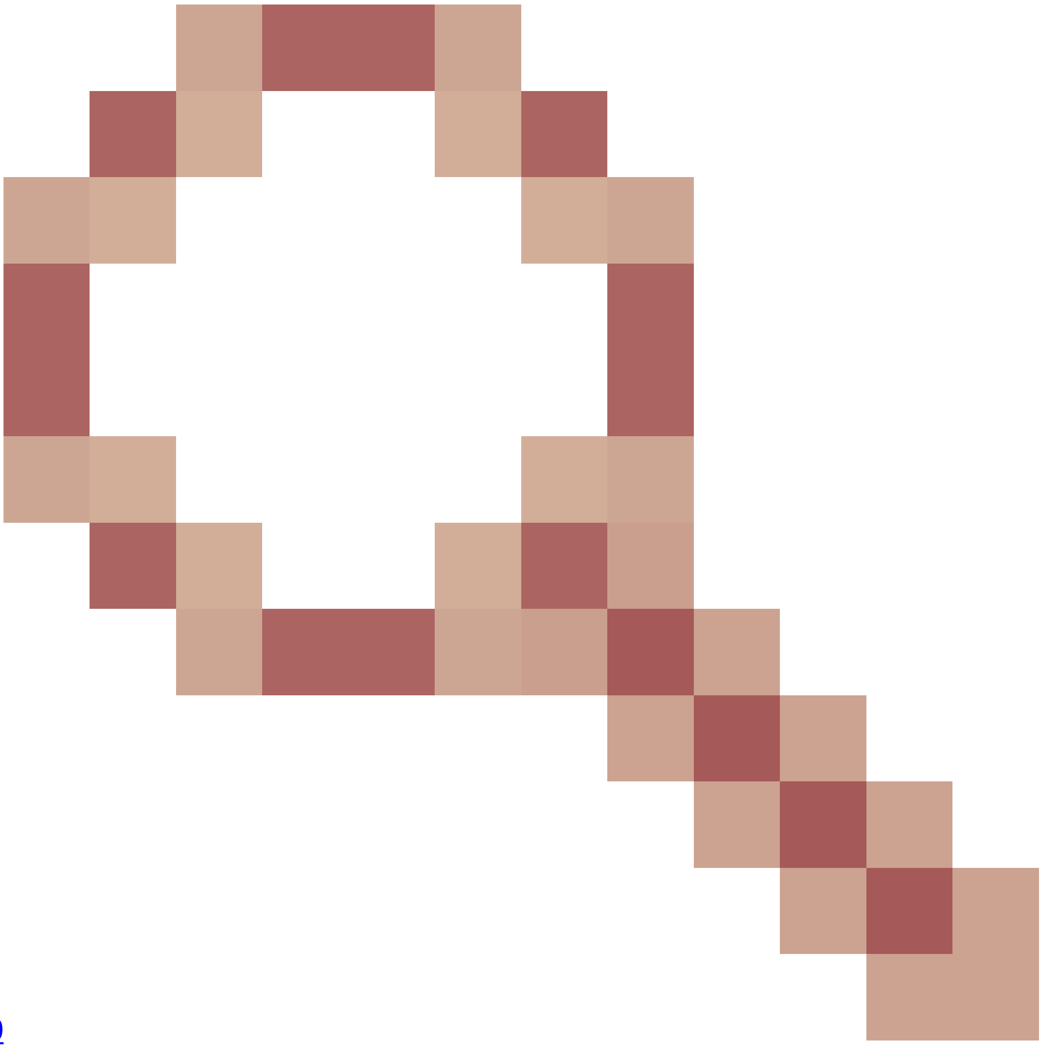


```
udp.port == 5688

No.  Delta    Source              Destination         Length Info                                                                      Payload Type
 59  0.000000 SamsungE_08:4c:4a                       251 Probe Request, SN=577, FN=0, Flags=........C, SSID=testewlcwlan
 60  0.003416 Cisco_36:4d:a0                          292 Probe Response, SN=410, FN=0, Flags=........C, BI=100, SSID=testewlcwlan
 61  0.038249 SamsungE_08:4c:4a                       107 Authentication, SN=578, FN=0, Flags=........C
 62  0.000001 Cisco_36:4d:a0                          107 Authentication, SN=411, FN=0, Flags=........C
 63  0.000000 SamsungE_08:4c:4a                       262 Association Request, SN=579, FN=0, Flags=........C, SSID=testewlcwlan
 64  0.021832 Cisco_36:4d:a0                          322 Association Response, SN=412, FN=0, Flags=........C
 65  0.000003 Cisco_36:4d:a0                          221 K                                                                            MSDU
 66  0.030097 SamsungE_08:4c:4a                       342 K                                                                            MSDU
 67  0.010008 Cisco_36:4d:a0                          391 K                                                                            MSDU
 68  0.009881 SamsungE_08:4c:4a                       199 K                                                                            MSDU
 69  0.009985 SamsungE_08:4c:4a                       107 A
 70  0.030106 Cisco_36:4d:a0                           93 A
 72  0.009817 SamsungE_08:4c:4a                        99 A
 73  0.020078 Cisco_36:4d:a0                           99 A
 74  0.000000 ::                                      207 N                                                                            MSDU
 83  0.300024 0.0.0.0                                 463 D                                                                            MSDU
 87  0.020237 Cisco_36:4d:a0                           99 A
 88  0.000001 192.168.101.1                           428 D                                                                            MSDU
 89  0.000000 192.168.101.1                           428 D                                                                            MSDU
 90  0.000001 SamsungE_08:4c:4a                        99 A
 91  0.000000 0.0.0.0                                 475 D                                                                            MSDU
 93  0.029855 192.168.101.1                           428 D                                                                            MSDU
 94  0.000001 192.168.101.1                           428 D                                                                            MSDU
 97  0.279945 SamsungE_08:4c:4a                       163 W                                                                            MSDU
 98  0.009931 Cisco_b7:3f:42      SamsungE_08:4c:4a   146 1                                                                            MSDU
 99  0.029996 SamsuneE 08:4c:4a   Broadcast           239 P

> Frame 59: 251 bytes on wire (2008 bits), 251 bytes captured (2008 bits) on interfa
> Ethernet II, Src: Cisco_e1:3d:ec (70:db:98:e1:3d:ec), Dst: GoodWayI_b7:b8:af (00:5
> Internet Protocol Version 4, Src: 192.168.1.83, Dst: 192.168.1.100
> User Datagram Protocol, Src Port: 5000, Dst Port: 5688
> AiroPeek/OmniPeek encapsulated IEEE 802.11
> 802.11 radio information
> IEEE 802.11 Probe Request, Flags: ........C
> IEEE 802.11 wireless LAN
```

Field	Value	Type	Default	Current
UDP port	5000	Integer, base 10	TAPA	PEEKREMOTE
UDP port	5688	Integer, base 10	SIGCOMP	PEEKREMOTE

```
wireshark_Ethernet_yellowCable_20200406150701_a16344.pcapng    Packets: 299 · Displayed: 168 (56.2%)    Profile: Default
```

List of bugs and enhancements around this feature :



[Cisco bug ID CSCvm09020](#)
DNS not seen by client trace anymore on 8.8

[Cisco bug ID CSCvm09015](#)

client trace shows many ICMP_other with null sequence number

Cisco bug ID CSCvm02676

AP COS client-trace does not capture webauth packets

Cisco Bug ID CSCvm02613

AP COS client-trace remote output does not work

Cisco Bug ID CSCvm00855

lient-trace SEQ numbers inconsistent

Control the AP Client trace from the 9800 WLC

You can configure several APs to do a radio client trace and trigger it from the

Step 1. Configure an AP trace profile that defines which traffic to capture

```
config term
   wireless profile ap trace <TRACE-NAME>
        filter all
        no filter probe
        output console-log
```

Step 2. Add the AP trace profile to an AP join profile that is used by the APs that you target.

```
ap profile < ap join profile name>
    trace <TRACE-NAME>
```

Ensure that this ap join profile is applied to a site tag that is used by your target APs

Step 4 Trigger start/stop

```
ap trace client start ap <ap name> client all/<mac>
ap trace client stop ap <ap name> client all/<mac>

ap trace client start site <site tag> client all/<mac>
ap trace client stop site <site tag> client all/<mac>
```

Verification commands :

```
show wireless profile ap trace summary
show wireless profile ap trace detailed PROF_NAME detail
sh ap trace client summary
show ap trace unsupported-ap summary
```

Client debug bundle on the AP

Rather than collecting a radio debug/capture, it may be easier to use the client debug bundle feature if you are debugging one or more specific clients.

Step 1. Identify which client you want to troubleshoot.

```
9164#show dot11 clients

Total dot11 clients: 6
       Client MAC   Slot ID WLAN ID AID WLAN Name RSSI  Maxrate is_wgb_wired is_
mld_sta
52:1E:34:C9:D6:F3         1       2  35   MySSID  -62      M7             No
     No
80:A9:97:2C:DC:6E         1       2  34   MySSID  -47 MCS112SS            No
     No
E8:D8:D1:1F:71:F3         0       2  35   MySSID  -62      M7             No
```

```
     No
6A:E4:06:E7:AB:E1         1       2  33   MySSID  -44 MCS112SS           No
     No
00:1D:63:70:AC:23         0       2  33   MySSID  -56      M7            No
     No
68:72:C3:FD:17:F5         0       2  34   MySSID  -53     M15            No
     No
```

Step 2. Start the debug for one or more client MAC address(es)

```
9164#debug client-bundle start debug 80:A9:97:2C:DC:6E
  WORD  <mac-addr2>
  <cr>
```

By default, nothing will be printed on screen. You can enable **terminal monitor** and see the debugs being printed live, but be aware that this wil make the terminal very hard to use. It's not required to print the debugs on the terminal in order to collect the bundle.

Step 3. You have to stop the debug bundle before you can upload the output of it :

```
debug client-bundle start debug 80:A9:97:2C:DC:6E
```

Step 4. Upload the bundle to an FTP or SCP server (as reminder the WLC can act as SCP server)

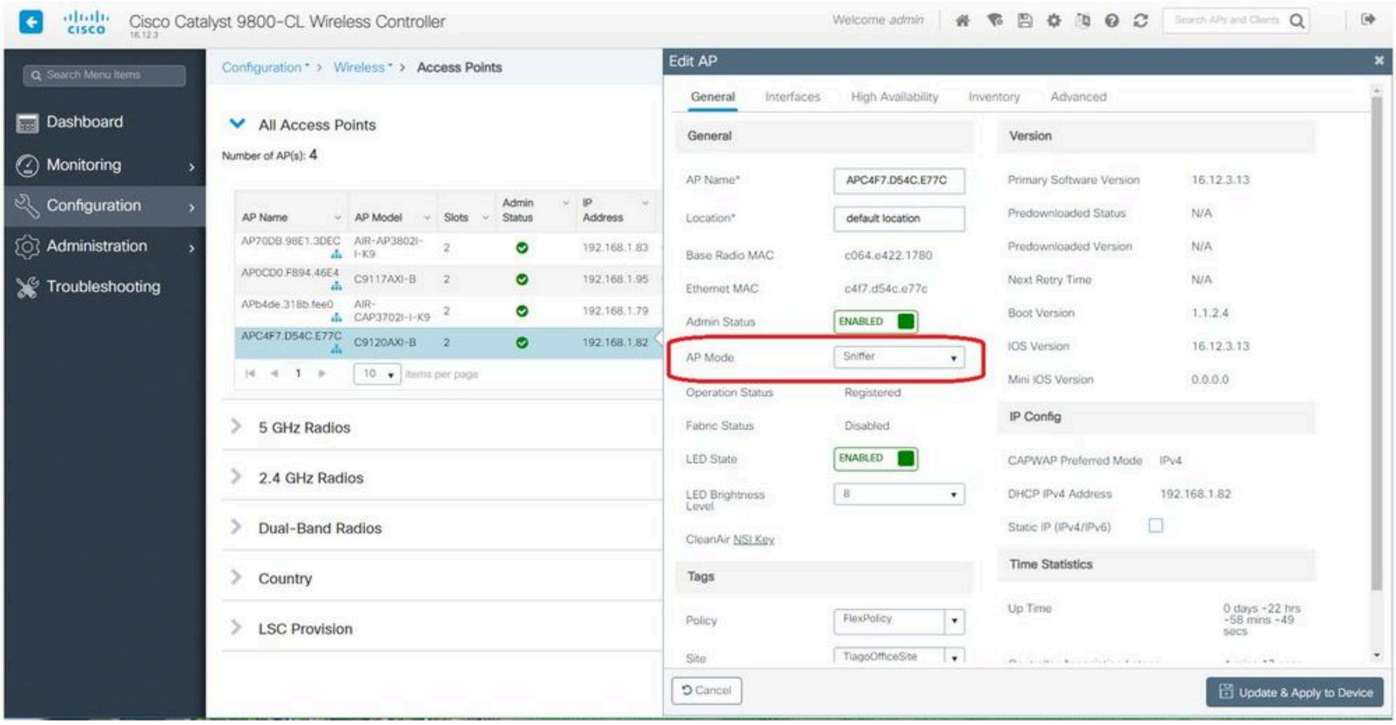
```
9164#debug client-bundle upload tftp 192.168.129.29 80:a9:97:2c:dc:6e
2024-09-04 11:58:48 Creating client bundle, please wait...

2024-09-04 11:59:01 Client bundle file 9164-_client_bundle.17.15.1.6.20240904.115848.tgz created.
2024-09-04 11:59:01 TFTP uploading...
Successful file transfer:
9164_client_bundle.17.15.1.6.20240904.115848.tgz

9164#
```

The TGZ bundle contains 4 files :

- 2 containing show commands pertaining to the radios and client
- 1 about the actual debugging (which is displayed on terminal if you do **term mon**)
- 1 containing syslogs

APs Catalyst 91xx in Sniffer Mode

The new Catalyst 9115, 9117, 9120 and 9130 can be configured in sniffer mode. The procedure is simular to previous AP models.



Cisco Catalyst 9800-CL Wireless Controller 16.12.3

Welcome *admin*

Configuration > Wireless > Access Points

All Access Points

Number of AP(s): 4

AP Name	AP Model	Slots	Admin Status	IP Address
AP70DB.98E1.3DEC	AIR-AP3802I-I-K9	2	✓	192.168.1.83
AP0CD0.F894.46E4	C9117AXI-B	2	✓	192.168.1.95
APb4de.318b.fee0	AIR-CAP3702I-I-K9	2	✓	192.168.1.79
APC4F7.D54C.E77C	C9120AXI-B	2	✓	192.168.1.82

5 GHz Radios

2.4 GHz Radios

Dual-Band Radios

Country

LSC Provision

Edit AP

General | Interfaces | High Availability | Inventory | Advanced

General

AP Name*	APC4F7.D54C.E77C
Location*	default location
Base Radio MAC	c064.e422.1780
Ethernet MAC	c4f7.d54c.e77c
Admin Status	ENABLED
AP Mode	Sniffer
Operation Status	Registered
Fabric Status	Disabled
LED State	ENABLED
LED Brightness Level	8
CleanAir NSI Key	

Tags

Policy	FlexPolicy
Site	TiagoOfficeSite

Version

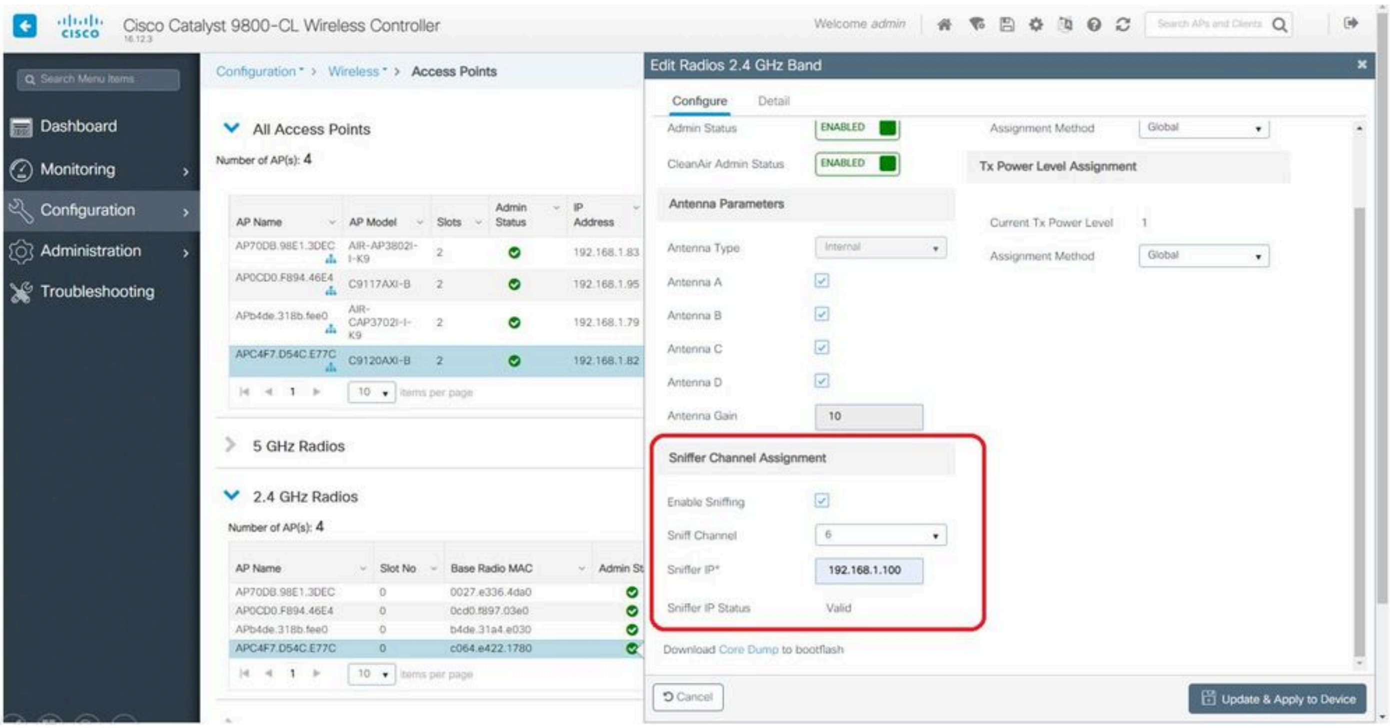
Primary Software Version	16.12.3.13
Predownloaded Status	N/A
Predownloaded Version	N/A
Next Retry Time	N/A
Boot Version	1.1.2.4
IOS Version	16.12.3.13
Mini IOS Version	0.0.0.0

IP Config

CAPWAP Preferred Mode	IPv4
DHCP IPv4 Address	192.168.1.82
Static IP (IPv4/IPv6)	□

Time Statistics

Up Time	0 days -22 hrs -58 mins -49 secs



Cisco Catalyst 9800-CL Wireless Controller 16.12.3

Welcome *admin*

Configuration > Wireless > Access Points

All Access Points

Number of AP(s): 4

AP Name	AP Model	Slots	Admin Status	IP Address
AP70DB.98E1.3DEC	AIR-AP3802I-I-K9	2	✓	192.168.1.83
AP0CD0.F894.46E4	C9117AXI-B	2	✓	192.168.1.95
APb4de.318b.fee0	AIR-CAP3702I-I-K9	2	✓	192.168.1.79
APC4F7.D54C.E77C	C9120AXI-B	2	✓	192.168.1.82

5 GHz Radios

2.4 GHz Radios

Number of AP(s): 4

AP Name	Slot No	Base Radio MAC	Admin St
AP70DB.98E1.3DEC	0	0027.e336.4da0	✓
AP0CD0.F894.46E4	0	0cd0.f897.03e0	✓
APb4de.318b.fee0	0	b4de.31a4.e030	✓
APC4F7.D54C.E77C	0	c064.e422.1780	✓

Edit Radios 2.4 GHz Band

Configure | Detail

Admin Status	ENABLED
CleanAir Admin Status	ENABLED

Antenna Parameters

Antenna Type	Internal
Antenna A	☑
Antenna B	☑
Antenna C	☑
Antenna D	☑
Antenna Gain	10

Sniffer Channel Assignment

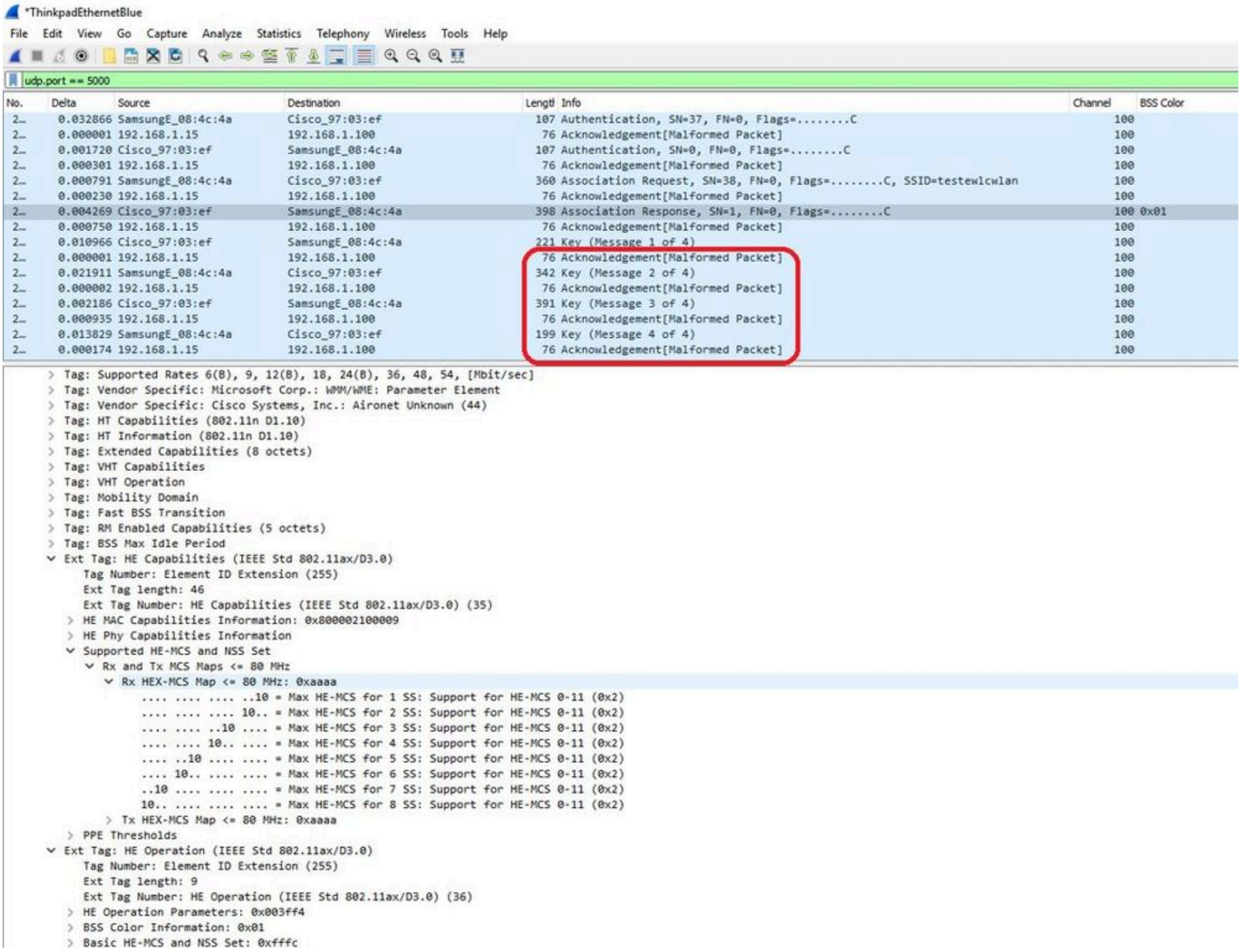
Enable Sniffing	☑
Sniff Channel	6
Sniffer IP*	192.168.1.100
Sniffer IP Status	Valid

Download Core Dump to bootflash

Assignment Method	Global

Tx Power Level Assignment

Current Tx Power Level	1
Assignment Method	Global



No.	Delta	Source	Destination	Length	Info	Channel	BSS Color
2…	0.032866	SamsungE_08:4c:4a	Cisco_97:03:ef	107	Authentication, SN=37, FN=0, Flags=........C	100	
2…	0.000001	192.168.1.15	192.168.1.100	76	Acknowledgement[Malformed Packet]	100	
2…	0.001720	Cisco_97:03:ef	SamsungE_08:4c:4a	107	Authentication, SN=0, FN=0, Flags=........C	100	
2…	0.000301	192.168.1.15	192.168.1.100	76	Acknowledgement[Malformed Packet]	100	
2…	0.000791	SamsungE_08:4c:4a	Cisco_97:03:ef	360	Association Request, SN=38, FN=0, Flags=........C, SSID=testewlcwlan	100	
2…	0.000230	192.168.1.15	192.168.1.100	76	Acknowledgement[Malformed Packet]	100	
2…	0.004269	Cisco_97:03:ef	SamsungE_08:4c:4a	398	Association Response, SN=1, FN=0, Flags=........C	100	0x01
2…	0.000750	192.168.1.15	192.168.1.100	76	Acknowledgement[Malformed Packet]	100	
2…	0.010966	Cisco_97:03:ef	SamsungE_08:4c:4a	221	Key (Message 1 of 4)	100	
2…	0.000001	192.168.1.15	192.168.1.100	76	Acknowledgement[Malformed Packet]	100	
2…	0.021911	SamsungE_08:4c:4a	Cisco_97:03:ef	342	Key (Message 2 of 4)	100	
2…	0.000002	192.168.1.15	192.168.1.100	76	Acknowledgement[Malformed Packet]	100	
2…	0.002186	Cisco_97:03:ef	SamsungE_08:4c:4a	391	Key (Message 3 of 4)	100	
2…	0.000935	192.168.1.15	192.168.1.100	76	Acknowledgement[Malformed Packet]	100	
2…	0.013829	SamsungE_08:4c:4a	Cisco_97:03:ef	199	Key (Message 4 of 4)	100	
2…	0.000174	192.168.1.15	192.168.1.100	76	Acknowledgement[Malformed Packet]	100	

```
> Tag: Supported Rates 6(B), 9, 12(B), 18, 24(B), 36, 48, 54, [Mbit/sec]
> Tag: Vendor Specific: Microsoft Corp.: WMM/WME: Parameter Element
> Tag: Vendor Specific: Cisco Systems, Inc.: Aironet Unknown (44)
> Tag: HT Capabilities (802.11n D1.10)
> Tag: HT Information (802.11n D1.10)
> Tag: Extended Capabilities (8 octets)
> Tag: VHT Capabilities
> Tag: VHT Operation
> Tag: Mobility Domain
> Tag: Fast BSS Transition
> Tag: RM Enabled Capabilities (5 octets)
> Tag: BSS Max Idle Period
v Ext Tag: HE Capabilities (IEEE Std 802.11ax/D3.0)
     Tag Number: Element ID Extension (255)
     Ext Tag length: 46
     Ext Tag Number: HE Capabilities (IEEE Std 802.11ax/D3.0) (35)
   > HE MAC Capabilities Information: 0x800002100009
   > HE Phy Capabilities Information
   v Supported HE-MCS and NSS Set
      v Rx and Tx MCS Maps <= 80 MHz
         v Rx HEX-MCS Map <= 80 MHz: 0xaaaa
              .... .... .... ..10 = Max HE-MCS for 1 SS: Support for HE-MCS 0-11 (0x2)
              .... .... .... 10.. = Max HE-MCS for 2 SS: Support for HE-MCS 0-11 (0x2)
              .... .... ..10 .... = Max HE-MCS for 3 SS: Support for HE-MCS 0-11 (0x2)
              .... .... 10.. .... = Max HE-MCS for 4 SS: Support for HE-MCS 0-11 (0x2)
              .... ..10 .... .... = Max HE-MCS for 5 SS: Support for HE-MCS 0-11 (0x2)
              .... 10.. .... .... = Max HE-MCS for 6 SS: Support for HE-MCS 0-11 (0x2)
              ..10 .... .... .... = Max HE-MCS for 7 SS: Support for HE-MCS 0-11 (0x2)
              10.. .... .... .... = Max HE-MCS for 8 SS: Support for HE-MCS 0-11 (0x2)
         > Tx HEX-MCS Map <= 80 MHz: 0xaaaa
   > PPE Thresholds
v Ext Tag: HE Operation (IEEE Std 802.11ax/D3.0)
     Tag Number: Element ID Extension (255)
     Ext Tag length: 9
     Ext Tag Number: HE Operation (IEEE Std 802.11ax/D3.0) (36)
   > HE Operation Parameters: 0x003ff4
   > BSS Color Information: 0x01
   > Basic HE-MCS and NSS Set: 0xfffc
```

Note: Data frames sent at WIFI 6 data rates are captured but, because peekremote is not up to date on Wireshark, they show as 802.11ax phy type as of now. The fix is in Wireshark 3.2.4 where Wireshark displays the proper wifi6 phy rate.

Note: Cisco APs can't capture MU-OFDMA frames at this time but can capture the trigger frames (sent at management data rate) that announce a MU-OFDMA window. You can already infer that MU-OFDMA happens (or not) and with which client.

Troubleshooting Tips

Path MTU

Although Path MTU discovery finds the optimal MTU for the AP, it is possible to override this settings manually.

On AireOS 8.10.130 WLC, the command **config ap pmtu disable <ap/all>** sets a static MTU for one or all APs rather than to rely on the dynamic discovery mechanism.

To enable debugs at boot time

You can run config boot debug capwap to enable capwap,DTLS and DHCP debugs at the next boot time, even before the OS has booted and the prompt is shown.

You also have "config boot debug memory xxxx" for several memory debugs.

You can see if boot debugs are enabled or not at next reboot with "show boot".

They can be disabled with the addition of the disable keyword at the end such as "config boot debug capwap disable".

Power save mechanism

The power save of a given client can be troubleshot by running

debug client trace <mac address>

Clients QoS

To verify that QoS tags are applied, you can run "***debug capwap client qos***".

It displays the UP value of packets for wireless clients.

It is not mac filterable as of 8.8 ; enhancement request Cisco bug IDCSCvm08899).

```
labAP#debug capwap client qos

[*08/20/2018 09:43:36.3171] chatter: set_qos_up :: SetQosPriority: bridged packet dst: 00:AE:FA:78:36:8
[*08/20/2018 09:43:45.0051] chatter: set_qos_up :: SetQosPriority: bridged packet dst: 00:AE:FA:78:36:8
[*08/20/2018 09:43:45.5463] chatter: set_qos_up :: SetQosPriority: bridged packet dst: 00:AE:FA:78:36:8
[*08/20/2018 09:43:46.5687] chatter: set_qos_up :: SetQosPriority: bridged packet dst: AC:81:12:C7:CD:3
[*08/20/2018 09:43:47.0982] chatter: set_qos_up :: SetQosPriority: bridged packet dst: AC:81:12:C7:CD:3
```

You can also verify the Qos UP to DSCP table on the AP as well as total amount of packets marked, shaped and dropped by Qos:

```
LabAP#show dot11 qos
Qos Policy Maps (UPSTREAM)

no policymap
Qos Stats (UPSTREAM)

total packets:   0
dropped packets: 0
```

```
marked packets:  0
shaped packets:  0
policed packets: 0
copied packets:  0

DSCP TO DOT1P (UPSTREAM)

Default dscp2dot1p Table Value:
[0]->0 [1]->2 [2]->10 [3]->18 [4]->26 [5]->34 [6]->46 [7]->48
Active dscp2dot1p Table Value:
[0]->0 [1]->2 [2]->10 [3]->18 [4]->26 [5]->34 [6]->46 [7]->48

Qos Policy Maps (DOWNSTREAM)

no policymap
Qos Stats (DOWNSTREAM)

total packets:   0
dropped packets: 0
marked packets:  0
shaped packets:  0
policed packets: 0
copied packets:  0

DSCP TO DOT1P (DOWNSTREAM)

Default dscp2dot1p Table Value:
[0]->0 [1]->-1 [2]->1 [3]->-1 [4]->1 [5]->-1 [6]->1 [7]->-1
[8]->-1 [9]->-1 [10]->2 [11]->-1 [12]->2 [13]->-1 [14]->2 [15]->-1
[16]->-1 [17]->-1 [18]->3 [19]->-1 [20]->3 [21]->-1 [22]->3 [23]->-1
[24]->-1 [25]->-1 [26]->4 [27]->-1 [28]->-1 [29]->-1 [30]->-1 [31]->-1
[32]->-1 [33]->-1 [34]->5 [35]->-1 [36]->-1 [37]->-1 [38]->-1 [39]->-1
[40]->-1 [41]->-1 [42]->-1 [43]->-1 [44]->-1 [45]->-1 [46]->6 [47]->-1
[48]->7 [49]->-1 [50]->-1 [51]->-1 [52]->-1 [53]->-1 [54]->-1 [55]->-1
[56]->7 [57]->-1 [58]->-1 [59]->-1 [60]->-1 [61]->-1 [62]->-1 [63]->-1
Active dscp2dot1p Table Value:
[0]->0 [1]->-1 [2]->1 [3]->-1 [4]->1 [5]->-1 [6]->1 [7]->-1
[8]->-1 [9]->-1 [10]->2 [11]->-1 [12]->2 [13]->-1 [14]->2 [15]->-1
[16]->-1 [17]->-1 [18]->3 [19]->-1 [20]->3 [21]->-1 [22]->3 [23]->-1
[24]->-1 [25]->-1 [26]->4 [27]->-1 [28]->-1 [29]->-1 [30]->-1 [31]->-1
[32]->-1 [33]->-1 [34]->5 [35]->-1 [36]->-1 [37]->-1 [38]->-1 [39]->-1
[40]->-1 [41]->-1 [42]->-1 [43]->-1 [44]->-1 [45]->-1 [46]->6 [47]->-1
[48]->7 [49]->-1 [50]->-1 [51]->-1 [52]->-1 [53]->-1 [54]->-1 [55]->-1
[56]->7 [57]->-1 [58]->-1 [59]->-1 [60]->-1 [61]->-1 [62]->-1 [63]->-1
LabAP#
```

When Qos policies are defined on the WLC and downloaded on the Flexconnect AP, you can verify them with :

```
AP780C-F085-49E6#show policy-map
2 policymaps
Policy Map BWLimitAAAClients            type:qos client:default
    Class BWLimitAAAClients_AVC_UI_CLASS
      drop

    Class BWLimitAAAClients_ADV_UI_CLASS
      set dscp af41 (34)
    Class class-default
      police rate 5000000 bps (625000Bytes/s)
        conform-action
        exceed-action


Policy Map platinum-up          type:qos client:default
    Class cm-dscp-set1-for-up-4
      set dscp af41 (34)


    Class cm-dscp-set2-for-up-4
      set dscp af41 (34)


    Class cm-dscp-for-up-5
      set dscp af41 (34)


    Class cm-dscp-for-up-6
      set dscp ef (46)


    Class cm-dscp-for-up-7
      set dscp ef (46)


    Class class-default
      no actions
```

In case of Qos rate-limiting :

```
AP780C-F085-49E6#show rate-limit client
Config:
              mac vap rt_rate_out rt_rate_in rt_burst_out rt_burst_in nrt_rate_out nrt_rate_in nrt_burs
A8:DB:03:6F:7A:46   2           0          0            0           0            0           0
Statistics:
            name    up  down
        Unshaped     0     0
  Client RT pass     0     0
 Client NRT pass     0     0
 Client RT drops     0     0
Client NRT drops     0 38621
               9 54922     0
```

Off-Channel scan

Debugging the off-channel scan of the AP can be useful when troubleshooting rogue detection (to validate if and when the AP goes on a specific channel to scan), but can also be useful in video troubleshoot where a sensitive real-time stream gets constant interruptions if the "off channel scan defer" feature is not used.

```
debug rrm off-channel defer
debug rrm off-chanel dbg (starting 17.8.1)
debug rrm off-channel schedule
debug rrm off-channel voice (starting 17.8.1)
debug rrm schedule (starting 17.8.1, debug NDP packet tx)
show trace dot_11 channel enable

[*06/11/2020 09:45:38.9530] wcp/rrm_userspace_0/rrm_schedule :: RRMSchedule process_int_duration_timer_
[*06/11/2020 09:45:39.0550] noise measurement channel 5 noise 89
[*06/11/2020 09:45:43.5490] wcp/rrm_userspace_1/rrm_schedule :: RRMSchedule process_int_duration_timer_
[*06/11/2020 09:45:43.6570] noise measurement channel 140 noise 97
```

Client Connectivity

It is possible to list clients that have been deauthenticated by the access point with the last event timestamp:

```
LabAP#show dot11 clients deauth
               timestamp               mac vap reason_code
Mon Aug 20 09:50:59 2018 AC:BC:32:A4:2C:D3   9           4
Mon Aug 20 09:52:14 2018 00:AE:FA:78:36:89   9           4
Mon Aug 20 10:31:54 2018 00:AE:FA:78:36:89   9           4
```

In the previous output, the reason code is the deauthentication reason code as detailed in this link :

https://community.cisco.com:443/t5/wireless-mobility-knowledge-base/802-11-association-status-802-11-deauth-reason-codes/ta-p/3148055

The vap refers to the identifier of the WLAN inside the AP (which is different from the WLAN ID on the WLC !!!).

You can cross-relate it with other outputs detailed subsequently which always mentions the vap of associated clients.

You can see the list of VAP ids with "***show controllers Dot11Radio 0/1 wlan***".

When clients are still associated, you can get details on their connection with:

```
LabAP#show dot11 clients

Total dot11 clients: 1
       Client MAC Slot ID WLAN ID AID WLAN Name RSSI Maxrate WGB
00:AE:FA:78:36:89       1      10   1   TestSSID -25 MCS82SS  No
```

A lot more details can be obtained about the client entry with:

```
LabAP#show client summ

Radio Driver client Summary:
==============================
wifi0
[*08/20/2018 11:54:59.5340]
[*08/20/2018 11:54:59.5340] Total STA List Count 0
[*08/20/2018 11:54:59.5340] | NO|              MAC|STATE|
[*08/20/2018 11:54:59.5340] -----------------------------
wifi1
[*08/20/2018 11:54:59.5357]
[*08/20/2018 11:54:59.5357] Total STA List Count 1
[*08/20/2018 11:54:59.5357] | NO|              MAC|STATE|
[*08/20/2018 11:54:59.5357] -----------------------------
[*08/20/2018 11:54:59.5357] |  1| 0:ffffffae:fffffffa:78:36:ffffff89|    8|

Radio Driver Client AID List:
==============================
wifi0
[*08/20/2018 11:54:59.5415]
[*08/20/2018 11:54:59.5415] Total STA-ID List Count 0
[*08/20/2018 11:54:59.5415] | NO|              MAC|STA-ID|
[*08/20/2018 11:54:59.5415] -------------------------------
wifi1
[*08/20/2018 11:54:59.5431]
[*08/20/2018 11:54:59.5431] Total STA-ID List Count 1
[*08/20/2018 11:54:59.5431] | NO|              MAC|STA-ID|
[*08/20/2018 11:54:59.5432] -------------------------------
[*08/20/2018 11:54:59.5432] |  1| 0:ffffffae:fffffffa:78:36:ffffff89|     6|

WCP client Summary:
=====================

              mac radio vap aid state       encr Maxrate is_wgb_wired      wgb_mac_addr
00:AE:FA:78:36:89     1   9   1   FWD AES_CCM128 MCS82SS        false 00:00:00:00:00:00


NSS client Summary:
=====================
Current Count: 3
|       MAC       | OPAQUE |PRI POL|VLAN|BR|TN|QCF|BSS|RADID|MYMAC|
|F8:0B:CB:E4:7F:41|00000000|      3|   0| 1| 1|  0|  2|    3|    1|
|F8:0B:CB:E4:7F:40|00000000|      3|   0| 1| 1|  0|  2|    3|    1|
|00:AE:FA:78:36:89|00000003|      1|   0| 1| 1|  0|  9|    1|    0|

Datapath IPv4 client Summary:
===============================
               id vap   port         node tunnel               mac        seen_ip     hashed_ip sniff_a
00:AE:FA:78:36:89   9 apr1v9 192.0.2.13      - 00:AE:FA:78:36:89 192.168.68.209 10.228.153.45  5.990000

Datapath IPv6 client Summary:
===============================
client               mac                 seen_ip6 age      scope   port
     1 00:AE:FA:78:36:89 fe80::2ae:faff:fe78:3689  61 link-local apr1v9

Wired client Summary:
=======================
mac port state local_client detect_ago associated_ago tx_pkts tx_bytes rx_pkts rx_bytes
```

You can force the disconnection of a specific client with :

```
test dot11 client deauthenticate
```

Traffic counters can be obtained per-client with:

```
LabAP#show client statistics wireless 00:AE:FA:78:36:89
Client MAC address: 00:AE:FA:78:36:89
Tx Packets                    : 621
Tx Management Packets         : 6
Tx Control Packets            : 153
Tx Data Packets               : 462
Tx Data Bytes                 : 145899
Tx Unicast Data Packets       : 600
Rx Packets                    : 2910
Rx Management Packets         : 13
Rx Control Packets            : 943
Rx Data Packets               : 1954
Rx Data Bytes                 : 145699
LabAP#
```

More on the radio level, a lot of information can be obtained in the "***show controllers***". When you add the client mac address, the supported data rates, current data reates, PHY capabilities as well as amount of retries and txfails, are displayed:

<#root>

```
LabAP#show controllers dot11Radio 0 client 00:AE:FA:78:36:89
              mac radio vap aid state       encr Maxrate is_wgb_wired      wgb_mac_addr
00:AE:FA:78:36:89     0   9   1   FWD AES_CCM128    M15         false 00:00:00:00:00:00
Configured rates for client 00:AE:FA:78:36:89
Legacy Rates(Mbps): 11
HT Rates(MCS):M0 M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 M12 M13 M14 M15
VHT Rates: 1SS:M0-7 2SS:M0-7

HT:yes     VHT:yes     HE:no     40MHz:no     80MHz:no     80+80MHz:no     160MHz:no
11w:no     MFP:no     11h:no     encrypt_polocy: 4
_wmm_enabled:yes     qos_capable:yes     WME(11e):no     WMM_MIXED_MODE:no
short_preamble:yes     short_slot_time:no     short_hdr:yes     SM_dyn:yes
short_GI_20M:yes     short_GI_40M:no     short_GI_80M:yes     LDPC:yes     AMSDU:yes     AMSDU_long:no
su_mimo_capable:yes     mu_mimo_capable:no     is_wgb_wired:no     is_wgb:no

Additional info for client 00:AE:FA:78:36:89
RSSI: -90
PS  : Legacy (Sleeping)
Tx Rate: 0 Kbps
Rx Rate: 117000 Kbps
VHT_TXMAP: 0
CCX Ver: 4

Statistics for client 00:AE:FA:78:36:89
              mac   intf TxData TxMgmt TxUC TxBytes
```

TxFail

```
 TxDcrd TxCumRetries RxData RxMgmt RxBytes RxErr TxRt   RxRt idle_counter stats_ago expiration
00:AE:FA:78:36:89 apr0v9      8      1    6    1038      1      0            0     31      1    1599

Per TID packet statistics for client 00:AE:FA:78:36:89
Priority Rx Pkts Tx Pkts Rx(last 5 s) Tx (last 5 s) QID Tx Drops Tx Cur Qlimit
```

```
       0      899     460           1              1 144        0      0   1024
       1        0       0           0              0 145        0      0   1024
       2        0       0           0              0 146        0      0   1024
       3       59       0           0              0 147        0      0   1024
       4        0       0           0              0 148        0      0   1024
       5        0       0           0              0 149        0      0   1024
       6        0       0           0              0 150        0      0   1024
       7        0       0           0              0 151        0      0   1024

Legacy Rate Statistics:
  (Mbps :   Rx, Tx, Tx-Retries)
11 Mbps :    2,    0,    0
 6 Mbps :    0,    9,    0

HT/VHT Rate Statistics:
(Rate/SS/Width :   Rx, Rx-Ampdu, Tx, Tx-Ampdu, Tx-Retries)
        0/1/20 :    4,    4,    0,    0,    0
        6/2/20 :    4,    4,    0,    0,    0
        7/2/20 :    5,    5,    0,    0,    0

webauth done:
false
```

In order to constantly keep track of a client data rate and/or RSSI value, you can run "**debug dot11 client rate address <mac>** " and this logs this information every second:

```
LabAP#debug dot11 client rate address 00:AE:FA:78:36:89
[*08/20/2018 14:17:28.0928]                 MAC   Tx-Pkts    Rx-Pkts  Tx-Rate  Rx-Rate   RSSI   SNR Tx-R
[*08/20/2018 14:17:28.0928] 00:AE:FA:78:36:89           0          0       12  a8.2-2s    -45    53
[*08/20/2018 14:17:29.0931] 00:AE:FA:78:36:89           7         18       12  a8.2-2s    -45    53
[*08/20/2018 14:17:30.0934] 00:AE:FA:78:36:89           3         18       12  a8.2-2s    -45    53
[*08/20/2018 14:17:31.0937] 00:AE:FA:78:36:89           2         20       12  a8.2-2s    -45    53
[*08/20/2018 14:17:32.0939] 00:AE:FA:78:36:89           2         20       12  a8.2-2s    -45    53
[*08/20/2018 14:17:33.0942] 00:AE:FA:78:36:89           2         21       12  a8.2-2s    -46    52
[*08/20/2018 14:17:34.0988] 00:AE:FA:78:36:89           1          4       12  a8.2-2s    -46    52
[*08/20/2018 14:17:35.0990] 00:AE:FA:78:36:89           9         23       12  a8.2-2s    -46    52
[*08/20/2018 14:17:36.0993] 00:AE:FA:78:36:89           3          7       12  a8.2-2s    -46    52
[*08/20/2018 14:17:37.0996] 00:AE:FA:78:36:89           2          6       12  a8.2-2s    -46    52
[*08/20/2018 14:17:38.0999] 00:AE:FA:78:36:89           2         14       12  a8.2-2s    -46    52
[*08/20/2018 14:17:39.1002] 00:AE:FA:78:36:89           2         10       12  a8.2-2s    -46    52
[*08/20/2018 14:17:40.1004] 00:AE:FA:78:36:89           1          6       12  a8.2-2s    -46    52
[*08/20/2018 14:17:41.1007] 00:AE:FA:78:36:89           9         20       12  a8.2-2s    -46    52
[*08/20/2018 14:17:42.1010] 00:AE:FA:78:36:89           0          0       12  a8.2-2s    -46    52
[*08/20/2018 14:17:43.1013] 00:AE:FA:78:36:89           2          8       12  a8.2-2s    -46    52
[*08/20/2018 14:17:44.1015] 00:AE:FA:78:36:89           0          0       12  a8.2-2s    -46    52
[*08/20/2018 14:17:45.1018] 00:AE:FA:78:36:89           0          0       12  a8.2-2s    -46    52
[*08/20/2018 14:17:46.1021] 00:AE:FA:78:36:89           0          0       12  a8.2-2s    -46    52
[*08/20/2018 14:17:47.1024] 00:AE:FA:78:36:89           0          0       12  a8.2-2s    -46    52
[*08/20/2018 14:17:48.1026] 00:AE:FA:78:36:89           7         15       12  a8.2-2s    -46    52
[*08/20/2018 14:17:49.1029] 00:AE:FA:78:36:89           0          6       12  a8.2-2s    -46    52
[*08/20/2018 14:17:50.1032] 00:AE:FA:78:36:89           0          0       12  a8.2-2s    -46    52
[*08/20/2018 14:17:51.1035] 00:AE:FA:78:36:89           1          7       12  a8.2-2s    -46    52
[*08/20/2018 14:17:52.1037] 00:AE:FA:78:36:89           0         17       12  a8.2-2s    -46    52
[*08/20/2018 14:17:53.1040] 00:AE:FA:78:36:89           1         19       12  a8.2-2s    -46    52
[*08/20/2018 14:17:54.1043] 00:AE:FA:78:36:89           2         17       12  a8.2-2s    -46    52
[*08/20/2018 14:17:55.1046] 00:AE:FA:78:36:89           2         22       12  a8.2-2s    -45    53
[*08/20/2018 14:17:56.1048] 00:AE:FA:78:36:89           1         18       12  a8.2-2s    -45    53
[*08/20/2018 14:17:57.1053] 00:AE:FA:78:36:89           2         18       12  a8.2-2s    -45    53
```

```
[*08/20/2018 14:17:58.1055] 00:AE:FA:78:36:89         12          37       12  a8.2-2s   -45     53
```

In this output, the Tx and Rx packet counters are packets transmitted in the second interval since it last printed, same thing for the Tx Retries. However the RSSI, SNR and data rate are the values from the last packet of that interval (and not an average for all packets in that interval).

Flexconnect scenarios

You can verify what ACLs are currently applied to a client in a pre-auth (CWA for example) or post-auth scenario:

```
AP#show client access-lists pre-auth all f48c.507a.b9ad
Pre-Auth URL ACLs for Client: F4:8C:50:7A:B9:AD
IPv4 ACL: IPv6 ACL:
ACTION URL-LIST

Resolved IPs for Client: F4:8C:50:7A:B9:AD
HIT-COUNT URL ACTION IP-LIST

REDIRECT
rule 0: allow true and ip proto 17 and src port 53
rule 1: allow true and ip proto 17 and dst port 53
rule 2: allow true and src 10.48.39.161mask 255.255.255.255
rule 3: allow true and dst 10.48.39.161mask 255.255.255.255
rule 4: deny true
No IPv6 ACL found


AP#show client access-lists post-auth all f48c.507a.b9ad
Post-Auth URL ACLs for Client: F4:8C:50:7A:B9:AD
IPv4 ACL: IPv6 ACL:
ACTION URL-LIST

Resolved IPs for Client: F4:8C:50:7A:B9:AD
HIT-COUNT URL ACTION IP-LIST

post-auth
rule 0: deny true and dst 192.0.0.0mask 255.0.0.0
rule 1: deny true and src 192.0.0.0mask 255.0.0.0
rule 2: allow true
No IPv6 ACL found
```

You can see the hit counters on the Flexconnect ACLs by enabling **debug flexconnect access-list counter client <client MAC>**

Subsequent executions of the **show client access-list pre-auth/post-auth all <MAC>** then add hit counters for each ACL entry. This works for all types of flex ACLs as of Cisco IOS® XE 17.13. In earlier versions, the same commands exist but only VLAN ACLs have their hit counters updated.

You can reset the ACL hit counters with **clear counters access-list client <mac>**

AP Filesystem

COS APs do not allow to list all the content of the file system as on unix platforms.

The command "***show filesystems***" gives a detail of the space usage and distribution on the current partition:

```
2802#show filesystems
Filesystem                Size      Used Available Use% Mounted on
/dev/ubivol/storage      57.5M    364.0K     54.1M   1% /storage
2802#
```

The command "***show flash***" lists the main files on the AP flash. You can also append the syslog or core keyword to list those specific folders.

```
ap_2802#show flash
Directory of /storage/
total 84
-rw-r--r--    1 root     root             0 May 21  2018 1111
-rw-r--r--    1 root     root             6 Apr 15 11:09 BOOT_COUNT
-rw-r--r--    1 root     root             6 Apr 15 11:09 BOOT_COUNT.reserve
-rw-r--r--    1 root     root            29 Apr 15 11:09 RELOADED_AT_UTC
drwxr-xr-x    2 root     root           160 Mar 27 13:53 ap-images
drwxr-xr-x    4 5        root          2016 Apr 15 11:10 application
-rw-r--r--    1 root     root          6383 Apr 26 09:32 base_capwap_cfg_info
-rw-r--r--    1 root     root            20 Apr 26 10:31 bigacl
-rw-r--r--    1 root     root          1230 Mar 27 13:53 bootloader.log
-rw-r--r--    1 root     root             5 Apr 26 09:29 bootloader_verify.shadow
-rw-r--r--    1 root     root            18 Jun 30  2017 config
-rw-r--r--    1 root     root          8116 Apr 26 09:32 config.flex
-rw-r--r--    1 root     root            21 Apr 26 09:32 config.flex.mgroup
-rw-r--r--    1 root     root             0 Apr 15 11:09 config.local
-rw-r--r--    1 root     root             0 Jul 26  2018 config.mesh.dhcp
-rw-r--r--    1 root     root           180 Apr 15 11:10 config.mobexp
-rw-r--r--    1 root     root             0 Jun  5  2018 config.oeap
-rw-r--r--    1 root     root          2253 Apr 26 09:43 config.wireless
drwxr-xr-x    2 root     root           160 Jun 30  2017 cores
drwxr-xr-x    2 root     root           320 Jun 30  2017 dropbear
drwxr-xr-x    2 root     root           160 Jun 30  2017 images
-rw-r--r--    1 root     root           222 Jan  2  2000 last_good_uplink_config
drwxr-xr-x    2 root     root           160 Jun 30  2017 lists
-rw-r--r--    1 root     root           215 Apr 16 11:01 part1_info.ver
-rw-r--r--    1 root     root           215 Apr 26 09:29 part2_info.ver
-rw-r--r--    1 root     root          4096 Apr 26 09:36 random_seed
-rw-r--r--    1 root     root             3 Jun 30  2017 rxtx_mode
-rw-r--r--    1 root     root            64 Apr 15 11:11 sensord_CSPRNG0
-rw-r--r--    1 root     root            64 Apr 15 11:11 sensord_CSPRNG1
drwxr-xr-x    3 support  root           224 Jun 30  2017 support
drwxr-xr-x    2 root     root          2176 Apr 15 11:10 syslogs
---------------------------------------------------------------------------
Filesystem                Size      Used Available Use% Mounted on
flash                    57.5M    372.0K     54.1M   1% /storage
```

Store and send syslogs

The syslog folder stores the syslog output from previous reboots. The command "***show log***" only shows syslog since the last reboot.

At each reboot cycle, the syslogs are written on incremental files.

```
artaki# show flash syslogs
Directory of /storage/syslogs/
total 128
-rw-r--r--    1 root     root         11963 Jul  6 15:23 1
-rw-r--r--    1 root     root         20406 Jan  1  2000 1.0
-rw-r--r--    1 root     root           313 Jul  6 15:23 1.last_write
-rw-r--r--    1 root     root         20364 Jan  1  2000 1.start
-rw-r--r--    1 root     root            33 Jul  6 15:23 1.watchdog_status
-rw-r--r--    1 root     root         19788 Jul  6 16:46 2
-rw-r--r--    1 root     root         20481 Jul  6 15:23 2.0
-rw-r--r--    1 root     root           313 Jul  6 16:46 2.last_write
-rw-r--r--    1 root     root         20422 Jul  6 15:23 2.start
---------------------------------------------------------------------------
Filesystem                Size      Used Available Use% Mounted on
flash                    57.6M     88.0K     54.5M   0% /storage

artaki# show flash cores
Directory of /storage/cores/
total 0
---------------------------------------------------------------------------
Filesystem                Size      Used Available Use% Mounted on
flash                    57.6M     88.0K     54.5M   0% /storage
```

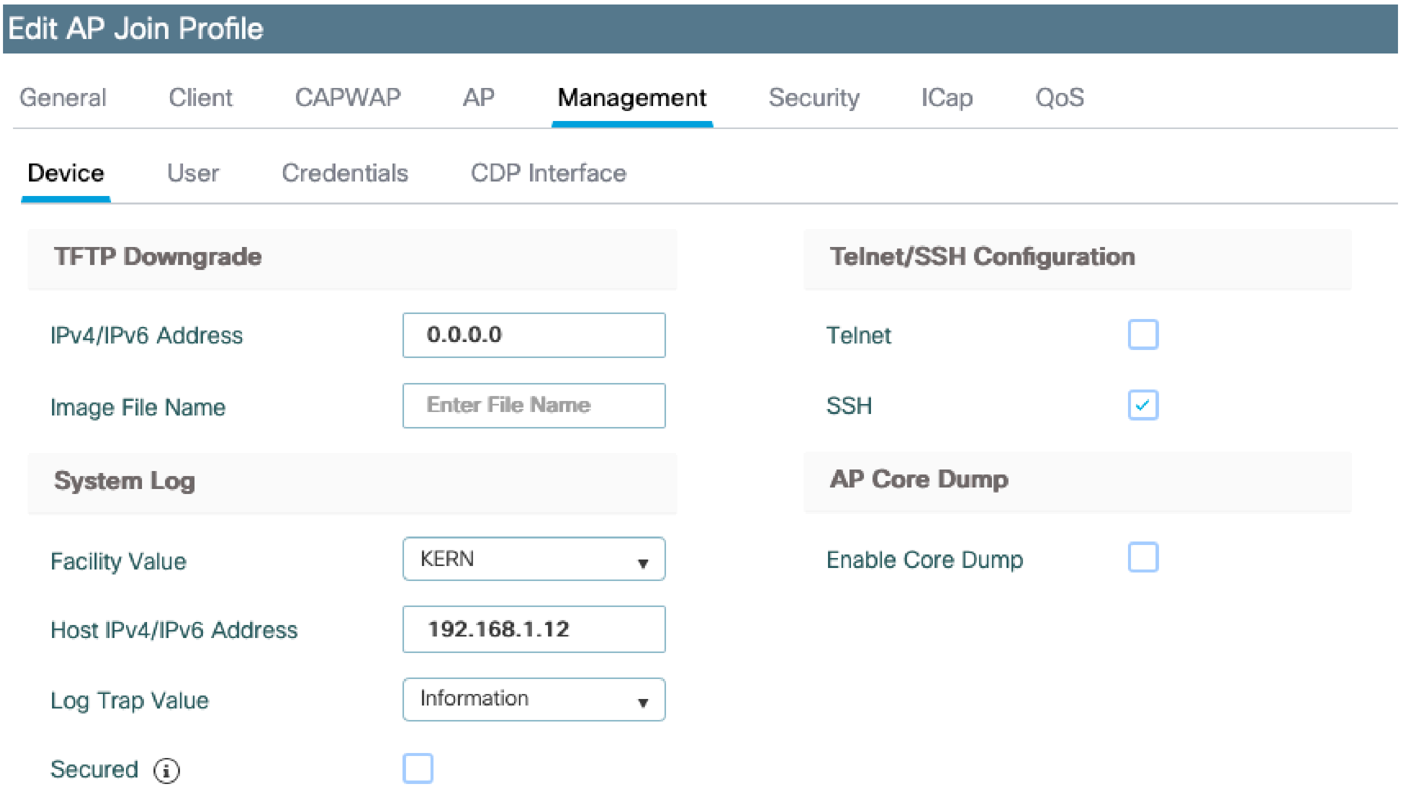
The first output after initial boot is file 1.0 and a file 1.1 is created if 1.0 becomes too long. After reboot, a new file 2.0 is created and so on.

From the WLC, you can configure the Syslog destination if you want your APs to send their syslog messages unicast to a specific server.

By default, APs send their syslogs to a broadcast address which can cause quite some broadcast storm, so ensure to configure a syslog server.

The AP sends via syslog by default whatever prints on its console output.

On 9800 Controller, you can change these parameters in the Configuration -> AP Join profile, under Management.



Edit AP Join Profile

TFTP Downgrade

IPv4/IPv6 Address	0.0.0.0
Image File Name	Enter File Name

System Log

Facility Value	KERN
Host IPv4/IPv6 Address	192.168.1.12
Log Trap Value	Information
Secured ⓘ	□

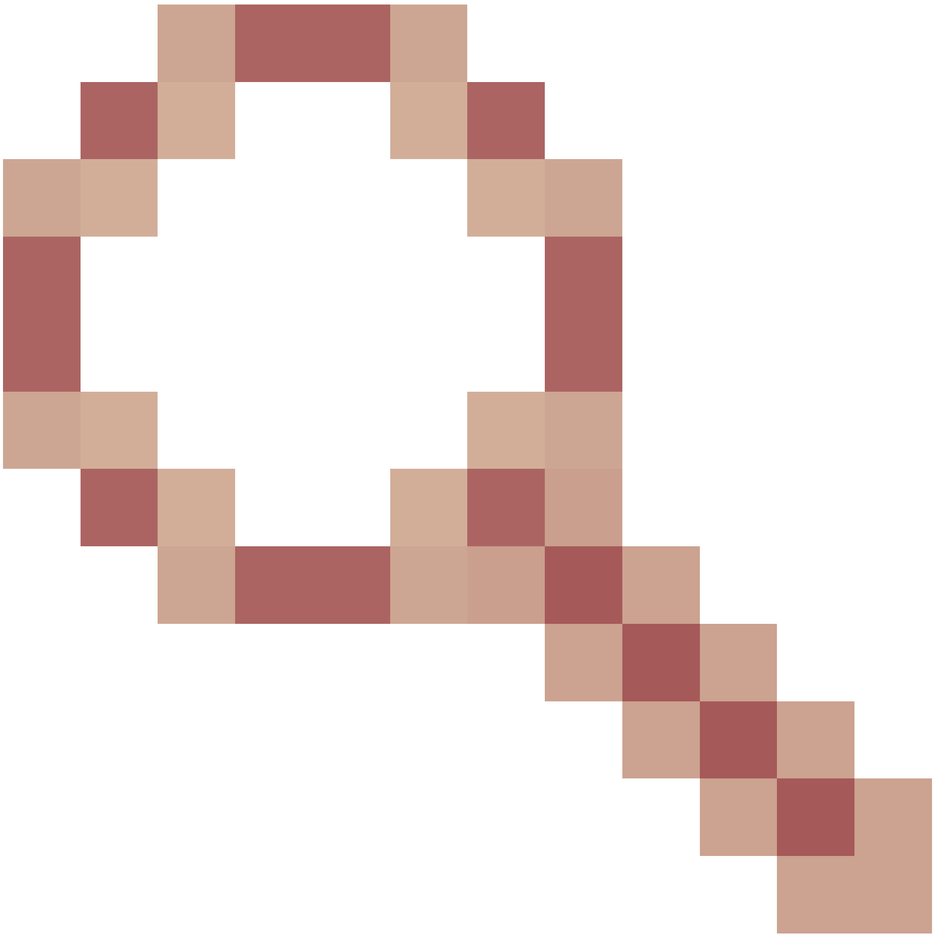
Telnet/SSH Configuration

Telnet	□
SSH	☑

AP Core Dump

Enable Core Dump	□

You can change the **Log Trap Value** to also send debugs via syslog. You can then enable debugs on the AP CLI and the output of these are sent via syslog messages to your configured server .



Due to Cisco Bug ID CSCvu75017

,only when you set the syslog facility to KERN (the default value) does the AP send syslog messages out.

If you are troubleshooting issues where an AP possibly loses network connectivity (or on a WGB for example), syslog is not as reliable as no messages are sent if the AP loses its uplink connectivity.

Therefore, reliance on the stored syslog files in flash is a great way to debug and store the output on the AP itself and then periodically upload it later on.

AP Support Bundle

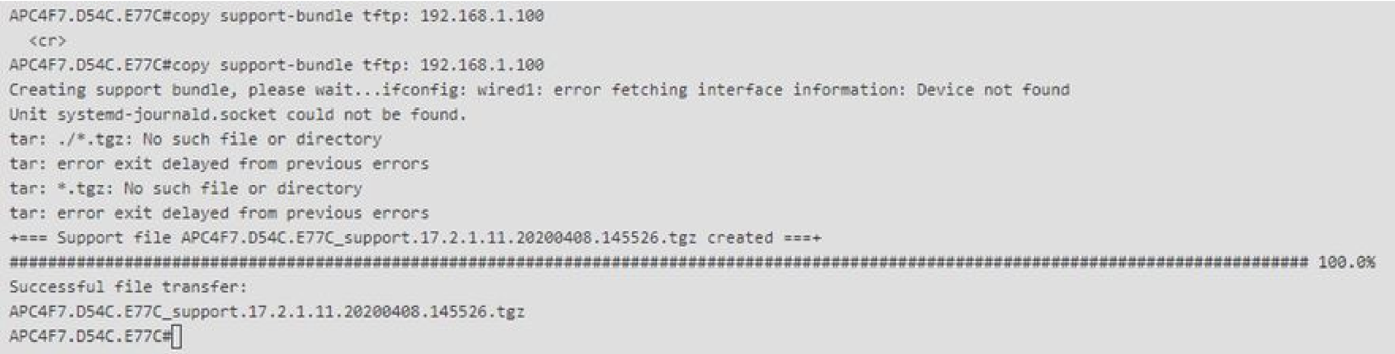
Some commonly collected diagnostic information of various types can be made available in a single bundle that you can upload from Access Points.

The diagnostic information that can you can include in the bundle are:

- AP show tech
- AP syslogs
- AP Capwapd Brain logs
- AP Startup & Message logs
- AP Coredump files

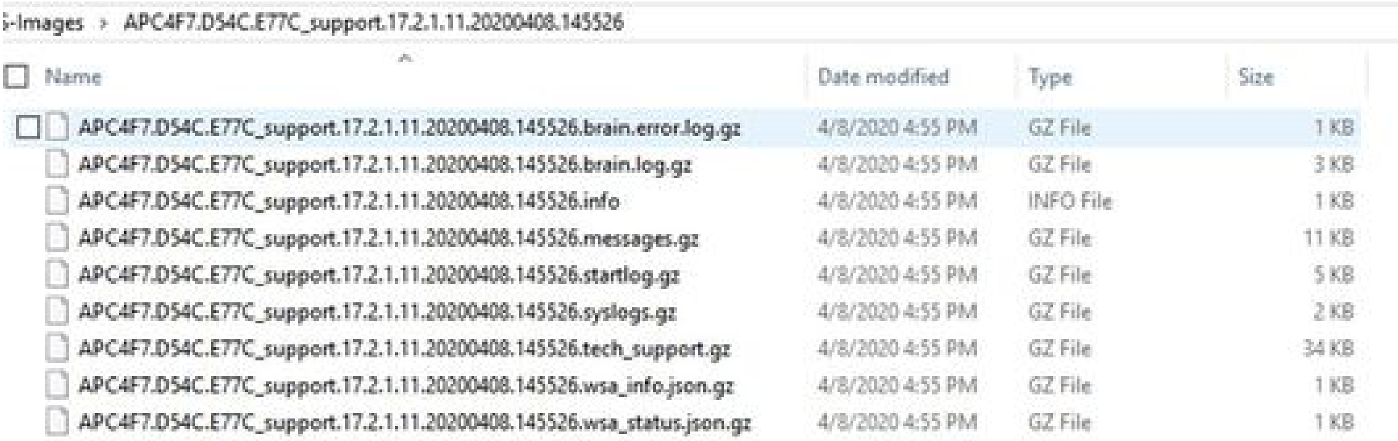
To get the AP support bundle you can go into the AP CLI and enter the command "***copy*** **support-bundle tftp: x.x.x.x**".

After this you can check for the file named with AP name appended with the **support.apversion.date.time.tgz** as shown subsequently :



```
APC4F7.D54C.E77C#copy support-bundle tftp: 192.168.1.100
  <cr>
APC4F7.D54C.E77C#copy support-bundle tftp: 192.168.1.100
Creating support bundle, please wait...ifconfig: wired1: error fetching interface information: Device not found
Unit systemd-journald.socket could not be found.
tar: ./*.tgz: No such file or directory
tar: error exit delayed from previous errors
tar: *.tgz: No such file or directory
tar: error exit delayed from previous errors
+=== Support file APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.tgz created ===+
########################################################################################################################################## 100.0%
Successful file transfer:
APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.tgz
APC4F7.D54C.E77C#
```

When you "untar" the file you can view the various files collected:



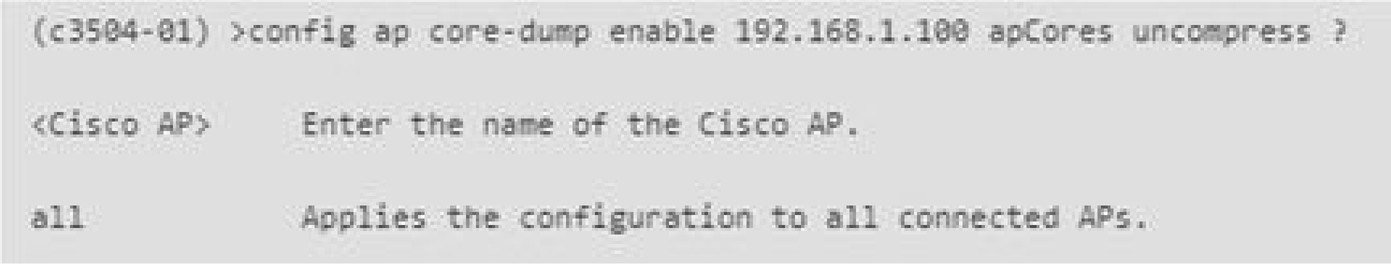
5-Images › APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526

□ Name	Date modified	Type	Size
□ APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.brain.error.log.gz	4/8/2020 4:55 PM	GZ File	1 KB
APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.brain.log.gz	4/8/2020 4:55 PM	GZ File	3 KB
APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.info	4/8/2020 4:55 PM	INFO File	1 KB
APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.messages.gz	4/8/2020 4:55 PM	GZ File	11 KB
APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.startlog.gz	4/8/2020 4:55 PM	GZ File	5 KB
APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.syslogs.gz	4/8/2020 4:55 PM	GZ File	2 KB
APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.tech_support.gz	4/8/2020 4:55 PM	GZ File	34 KB
APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.wsa_info.json.gz	4/8/2020 4:55 PM	GZ File	1 KB
APC4F7.D54C.E77C_support.17.2.1.11.20200408.145526.wsa_status.json.gz	4/8/2020 4:55 PM	GZ File	1 KB

Collect AP Core Files Remotely

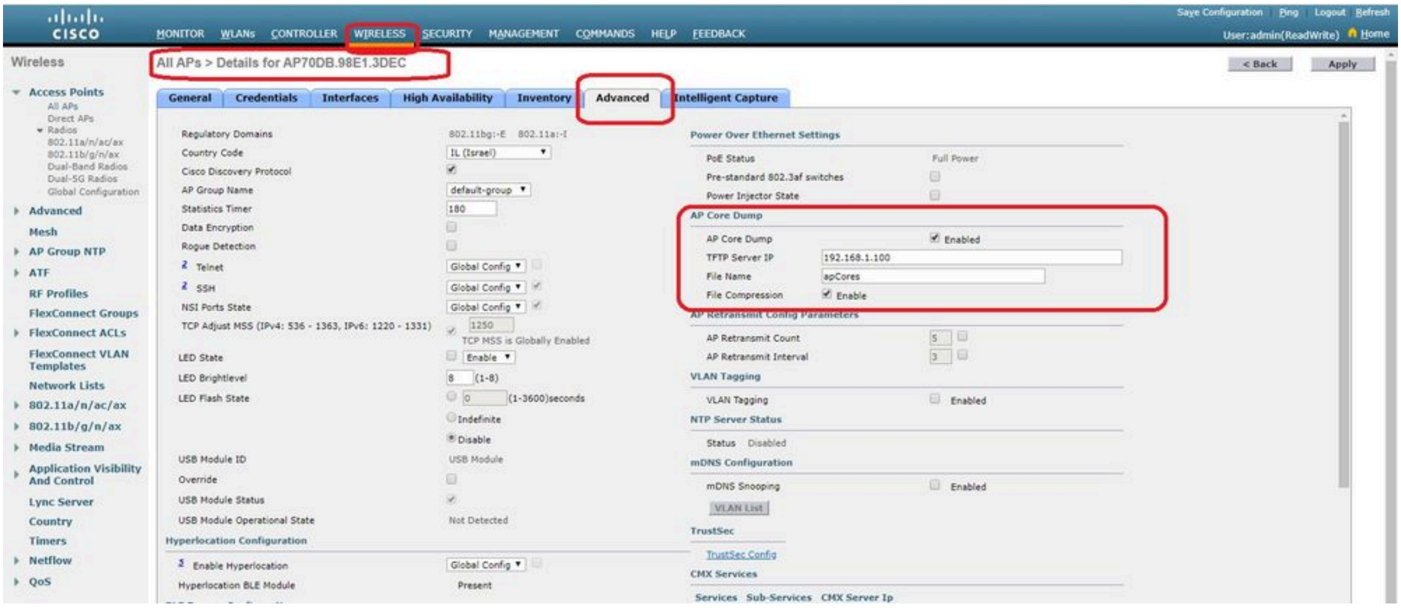
To collect AP core files remotely, please enable core dump to be included in support bundle and then Upload support bundle from the AP, or send directly to tftp server. The subsequent examples use tftp server 192.168.1.100.

AireOS CLI



```
(c3504-01) >config ap core-dump enable 192.168.1.100 apCores uncompress ?

<Cisco AP>      Enter the name of the Cisco AP.

all             Applies the configuration to all connected APs.
```

AireOS GUI



All APs > Details for AP70DB.98E1.3DEC

Regulatory Domains 802.11bg:-E 802.11a:-I
Country Code IL (Israel)
Cisco Discovery Protocol ☑
AP Group Name default-group
Statistics Timer 180
Data Encryption ☐
Rogue Detection ☐
Telnet Global Config ☐
SSH Global Config ☑
NSI Ports State Global Config ☑
TCP Adjust MSS (IPv4: 536 - 1363, IPv6: 1220 - 1331) ☑ 1250
TCP MSS is Globally Enabled
LED State ☐ Enable
LED Brightlevel 8 (1-8)
LED Flash State ○ 0 (1-3600)seconds
○ Indefinite
● Disable
USB Module ID USB Module
Override ☐
USB Module Status ☑
USB Module Operational State Not Detected

Hyperlocation Configuration
Enable Hyperlocation Global Config ☐
Hyperlocation BLE Module Present

Power Over Ethernet Settings
PoE Status Full Power
Pre-standard 802.3af switches ☐
Power Injector State ☐

AP Core Dump
AP Core Dump ☑ Enabled
TFTP Server IP 192.168.1.100
File Name apCores
File Compression ☑ Enable

AP Retransmit Config Parameters
AP Retransmit Count 5 ☐
AP Retransmit Interval 3 ☐

VLAN Tagging
VLAN Tagging ☐ Enabled

NTP Server Status
Status Disabled

mDNS Configuration
mDNS Snooping ☐ Enabled

TrustSec
TrustSec Config

CMX Services
Services Sub-Services CMX Server Ip

Cisco IOS® CLI

<#root>

```
eWLC-9800-01(
```

config

```
)#ap profile TiagoOffice
eWLC-9800-01(
```

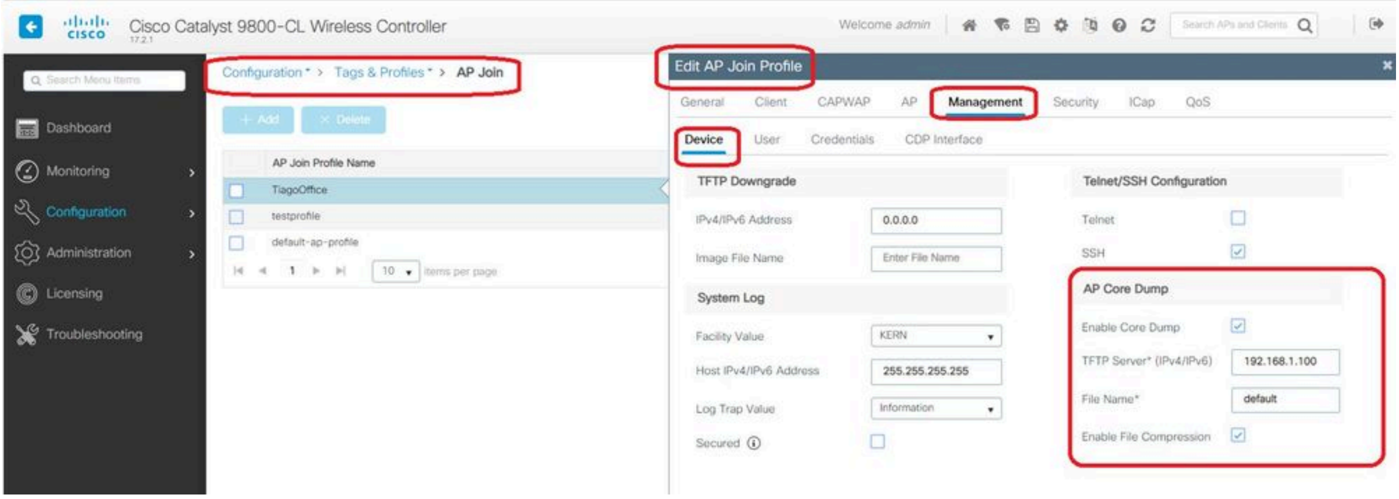
config-

ap

-profile

```
)#core-dump tftp-server 192.168.1.100 file apCores uncompress
```

Cisco IOS® GUI



Configuration > Tags & Profiles > AP Join

AP Join Profile Name
TiagoOffice
testprofile
default-ap-profile

Edit AP Join Profile

General | Client | CAPWAP | AP | **Management** | Security | ICap | QoS

Device | User | Credentials | CDP Interface

TFTP Downgrade
- IPv4/IPv6 Address: 0.0.0.0
- Image File Name: Enter File Name

System Log
- Facility Value: KERN
- Host IPv4/IPv6 Address: 255.255.255.255
- Log Trap Value: Information
- Secured: □

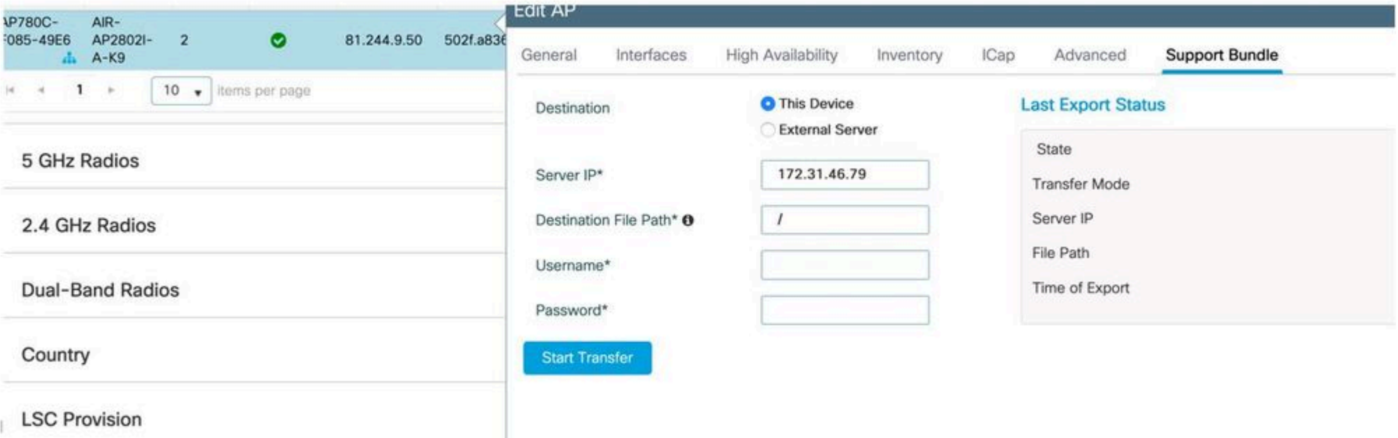
Telnet/SSH Configuration
- Telnet: □
- SSH: ☑

AP Core Dump
- Enable Core Dump: ☑
- TFTP Server* (IPv4/IPv6): 192.168.1.100
- File Name*: default
- Enable File Compression: ☑

As from Cisco IOS® XE 17.3.1, you have a Support Bundle tab and can download the AP SB from the WLC GUI.

All it does is execute “***copy support-bundle***” command on the AP and sends it via SCP to the WLC (because WLC can be an SCP server).

And then you can download it from your browser:



Edit AP

General | Interfaces | High Availability | Inventory | ICap | Advanced | **Support Bundle**

- Destination: ● This Device ○ External Server
- Server IP*: 172.31.46.79
- Destination File Path*: /
- Username*:
- Password*:

Last Export Status
- State
- Transfer Mode
- Server IP
- File Path
- Time of Export

This means you can manually do the same trick in eWLC releases before 17.3.1:

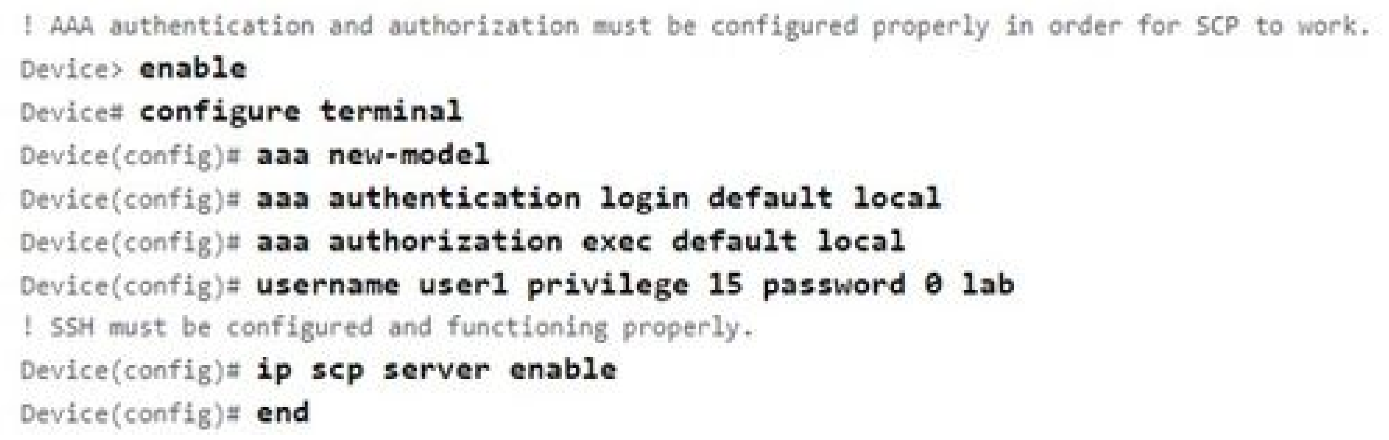
Copy the support bundle from AP via SCP to eWLC IP if you don’t have a TFTP server reachable to the AP.

The eWLC is usually reachable via SSH from the AP, so that’s a good trick for pre-17.3.

Step 1. Enable SSH on 9800 v17.2.1

Step 2. Enable SCP on Cisco IOS® XE v17.2.1

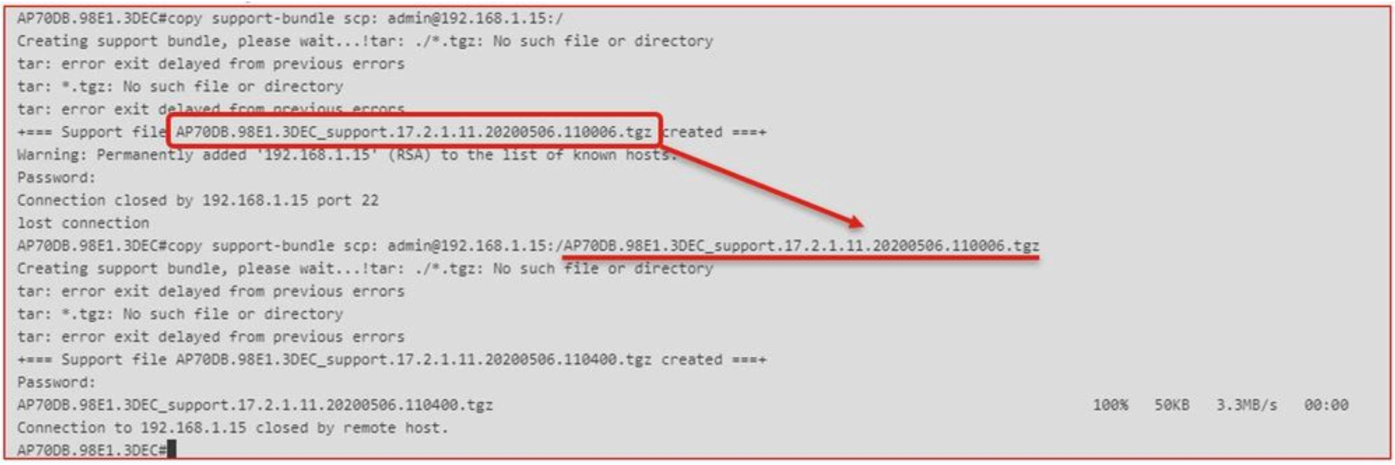
This example shows how to configure the server-side functionality of SCP. This example uses a locally defined username and password:



```
! AAA authentication and authorization must be configured properly in order for SCP to work.
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication login default local
Device(config)# aaa authorization exec default local
Device(config)# username user1 privilege 15 password 0 lab
! SSH must be configured and functioning properly.
Device(config)# ip scp server enable
Device(config)# end
```

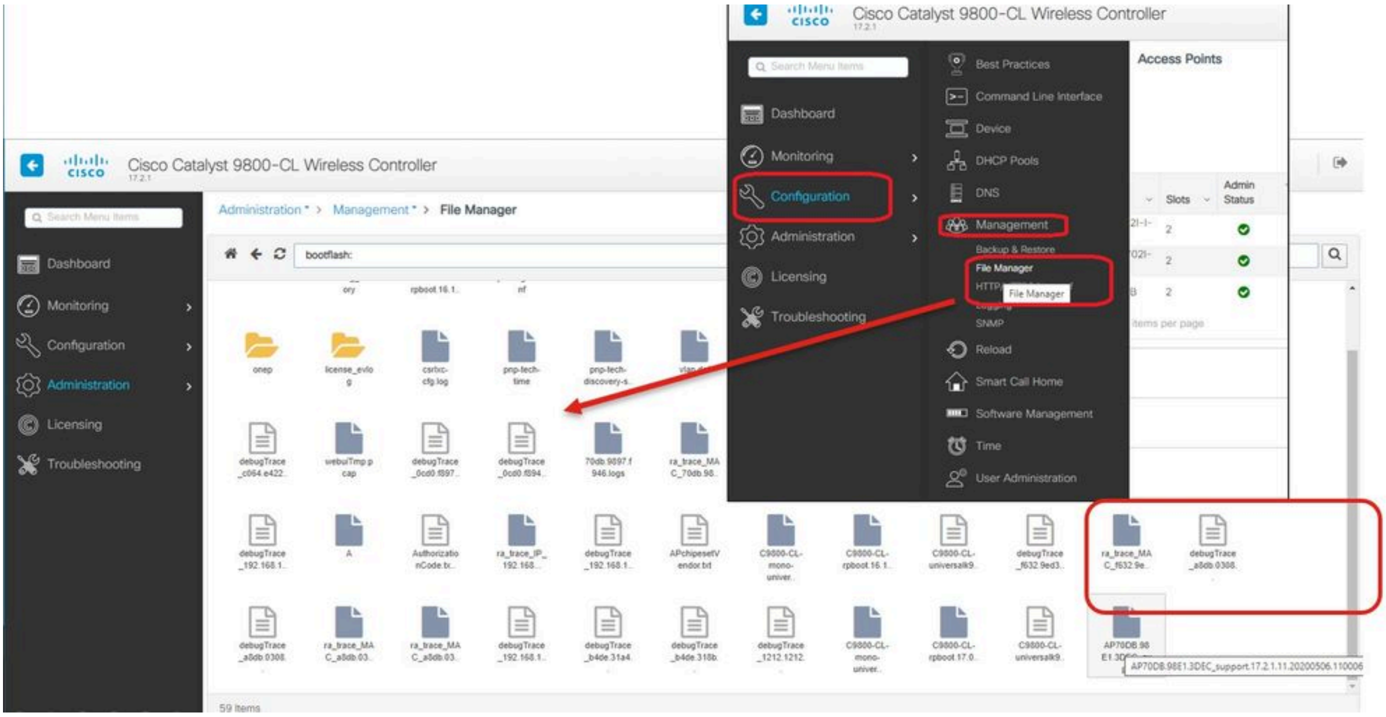
Step 3. Use the command “***copy support-bundle***” and we need to specify the filename to be created in the SCP server.

Tip: You can run the command once to get a meaningful filename, and then copy/paste that filename in the command:



```
AP70DB.98E1.3DEC#copy support-bundle scp: admin@192.168.1.15:/
Creating support bundle, please wait...!tar: ./*.tgz: No such file or directory
tar: error exit delayed from previous errors
tar: *.tgz: No such file or directory
tar: error exit delayed from previous errors
+=== Support file AP70DB.98E1.3DEC_support.17.2.1.11.20200506.110006.tgz created ===+
Warning: Permanently added '192.168.1.15' (RSA) to the list of known hosts.
Password:
Connection closed by 192.168.1.15 port 22
lost connection
AP70DB.98E1.3DEC#copy support-bundle scp: admin@192.168.1.15:/AP70DB.98E1.3DEC_support.17.2.1.11.20200506.110006.tgz
Creating support bundle, please wait...!tar: ./*.tgz: No such file or directory
tar: error exit delayed from previous errors
tar: *.tgz: No such file or directory
tar: error exit delayed from previous errors
+=== Support file AP70DB.98E1.3DEC_support.17.2.1.11.20200506.110400.tgz created ===+
Password:
AP70DB.98E1.3DEC_support.17.2.1.11.20200506.110400.tgz                                        100%   50KB   3.3MB/s   00:00
Connection to 192.168.1.15 closed by remote host.
AP70DB.98E1.3DEC#
```

Step 4. Then you can go into the eWLC GUI and get the file under: **Administration > Management > File Manager**:



IoT and Bluetooth

The gRPC server logs can be checked on the AP with :

```
AP# show grpc server log
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] spaces conn url 10.22.243.33:8000"
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] entering stopDNAspacesTmpTokenRoutine"
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] exiting stopDNAspacesTmpTokenRoutine"
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] entering startDNAspacesTmpTokenRoutine"
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] launching token request cycle"
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] exiting startDNAspacesTmpTokenRoutine"
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] spaces token expiration time 2020-04-02 01:36:52 +00
time="2020-04-01T01:36:52Z" level=info msg=" Calling startDNASpacesConn routine "
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] Receive Success status"
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] Connection not in ready state sleeping for 10 second
time="2020-04-01T01:37:02Z" level=info msg="[DNAS] Setup Stream for the gRPC connection"
time="2020-04-01T01:37:02Z" level=info msg="[DNAS] Connect RPC Succeeded."
time="2020-04-01T01:37:02Z" level=info msg="[DNAS] RX  routine got enabled "
time="2020-04-01T01:37:02Z" level=info msg="[DNAS] TX routine got enabled "
```

Connectivity to Cisco DNA Spaces connector can be verified with :

```
AP# show cloud connector key access
Token Valid : Yes
Token Stats :
        Number of Attempts  : 44
        Number of Failures  : 27
        Last Failure on     : 2020-03-28 02:02:15.649556818 +0000 UTC m=+5753.097022576
        Last Failure reason : curl: SSL connect error
        Last Success on     : 2020-04-01 00:48:37.313511596 +0000 UTC m=+346934.760976625
        Expiration time     : 2020-04-02 00:48:37 +0000 UTC
Connection Retry Interval : 30
```

```
AP# show cloud connector connection detail
Connection State           : READY
Connection Url             :  10.22.243.33:8000
Certificate Available      :  true
Controller Ip              :  10.22.243.31
Stream Setup Interval      :  30
Keepalive Interval         :  30
Last Keepalive Rcvd On     :  2020-04-01 00:32:47.891433113 +0000 UTC m=+345985.338898246
Number of Dials               :  2
Number of Tx Pkts             :  2788175
Number of Rx Pkts             :  11341
Number of Dropped Pkts        :  0
Number of Rx Keepalive        :  11341
Number of Tx Keepalive        :  11341
Number of Rx Cfg Request      :  0
Number of Tx AP Cfg Resp      :  0
Number of Tx APP Cfg Resp     :  0
Number of Tx APP state pkts   :  5
Number of Tx APP data pkts    :  2776829
```

To see the current BLE broadcasting config of the AP :

```
AP# show controllers ioTRadio ble 0 broadcast

BLE Profile Config
-------------------
Active profile           : v-iBeacon
Profile 0 (iBeacon)
UUID                     : 00001000000000000000000000000000
Interval (ms)            : 100
Power (dBm)              : -21
Advertised Power (dBm)   : -65
Minor                    : 0
Major                    : 0
TxPower byte             : bfbfbfbfbfbfbfbfbfbfbfbfbf

Profile 1 (Eddystone UID)
Namespace (hex)          : 0000000000005446089c
Instance-ID (hex)        : 7f0000001f00

Profile 2 (Eddystone URL)
URL                      : http://www.
```

To see the scanned results :

```
AP# show controllers ioTRadio ble 0 scan brief
      Profile               MAC  RSSI(-dBm)  RSSI@1meter(-dBm)        Last-heard
      Unknown 3C:1D:AF:62:EC:EC          88                  0 0000D:00H:00M:01S
      iBeacon 18:04:ED:04:1C:5F          86                 65 0000D:00H:00M:01S
      Unknown 18:04:ED:04:1C:5F          78                 65 0000D:00H:00M:01S
      Unknown 04:45:E5:28:8E:E7          85                 65 0000D:00H:00M:01S
      Unknown 2D:97:FA:0F:92:9A         91                65 0000D:00H:00M:01S
      iBeacon E0:7D:EA:16:35:35         68                65 0000D:00H:00M:01S
      Unknown E0:7D:EA:16:35:35         68                65 0000D:00H:00M:01S
      iBeacon 04:EE:03:53:74:22         45               256 0000D:00H:00M:01S
      Unknown 04:EE:03:53:74:22         45               256 0000D:00H:00M:01S
              04:EE:03:53:6A:3A         72               N/A 0000D:00H:00M:01S
      Unknown 04:EE:03:53:6A:3A         72                65 0000D:00H:00M:01S
      iBeacon E0:7D:EA:16:35:35         68                65 0000D:00H:00M:01S
      Unknown E0:7D:EA:16:35:35         67                65 0000D:00H:00M:01S
      iBeacon 04:EE:03:53:74:22         60               256 0000D:00H:00M:01S
      Unknown 04:EE:03:53:74:22         60               256 0000D:00H:00M:01S
Eddystone URL 04:EE:03:53:6A:3A         72               N/A 0000D:00H:00M:01S
```

When the AP acts in Advanced BLE gateway mode where an app is deployed, you can check the status of the IoX application with :

```
AP#show iox applications
Total Number of Apps : 1
--------------------------
App Name                     : cisco_dnas_ble_iox_app
   App Ip                    : 192.168.11.2
   App State                 : RUNNING
   App Token                 : 02fb3e98-ac02-4356-95ba-c43e8a1f4217
   App Protocol              : ble
   App Grpc Connection       : Up
   Rx Pkts From App          : 3878345
   Tx Pkts To App            : 6460
   Tx Pkts To Wlc            : 0
   Tx Data Pkts To DNASpaces : 3866864
   Tx Cfg Resp To DNASpaces  : 1
   Rx KeepAlive from App     : 11480
   Dropped Pkts              : 0
   App keepAlive Received On : Mar 24 05:56:49
```

You can connect to the IOX application with these commands and then monitor the logs during floor beacon configuration :

```
AP#connect iox application
/ #

/# tail -F /tmp/dnas_ble.log
Tue Mar 24 06:55:21 2020 [INFO]: Starting DNA Spaces BLE IOx Application
Tue Mar 24 06:55:21 2020 [INFO]: Auth token file contents: db26a8ab-e800-4fe9-a128-80683ea17b12
Tue Mar 24 06:55:21 2020 [INFO]: Setting gRPC endpoint to: 1.1.7.101:57777
Tue Mar 24 06:55:21 2020 [INFO]: Auth with token: db26a8ab-e800-4fe9-a128-80683ea17b12
Tue Mar 24 06:55:21 2020 [INFO]: Attempt to connect to DNAS Channel
Tue Mar 24 06:55:21 2020 [INFO]: Starting to run metrics
Tue Mar 24 06:55:21 2020 [INFO]: Starting to run Channel Keepalive
Tue Mar 24 06:55:21 2020 [INFO]: Initialize DNAS Reader Channel
Tue Mar 24 06:55:21 2020 [INFO]: Start listener for messages
Tue Mar 24 06:55:21 2020 [INFO]: Running BLE scan thread
```

Conclusion

There are many troubleshooting tools available to help us in the resolutions of problems related to COS APs.

This document lists the most commonly used ones and is regularly updated.