Troubleshoot Fragmentation Issues: Affecting c9800 Wireless Controller with Azure

Contents

Introduction

Symptoms

Error on ISE server

Detailed Log Analysis:

- Wireless controller EPC:
- ISE TCP Dumps
- Azure Side Capture with analysis:

Workaround suggested from Wireless controller end:

Solution:

Introduction

This document describes a known problem with the Azure platform leading to packet loss due to the mishandling of out-of-sequence fragments.

Symptoms

Affected Products: Catalyst 9800-CL Wireless Controller hosted on Azure or Identity Service Engine hosted on Azure.

SSID Setup: Configured for 802.1x EAP-TLS with central authentication.

Conduct : While utilizing the 9800-CL hosted on the Azure platform with an EAP-TLS based SSID you can encounter connectivity issues. The clients may encounter difficulties during the authentication phase.

Error on ISE server

Error code 5411 indicating that the supplicant has ceased communication with ISE during the EAP-TLS certificate exchange.

Detailed Log Analysis:

Here is an illustration of one of the impacted configurations: In the 9800 Wireless controller, the SSID is set up for 802.1x, and the AAA server is configured for EAP-TLS. When a client attempts authentication, particularly during the certificate exchange phase, the client sends a certificate that exceeds the maximum transmission unit (MTU) size on the Wireless controller. The 9800 Wireless controller then fragments this large packet and sends the fragments sequentially to AAA server. However, these fragments do not arrive in the correct order at the physical host, leading to packet drop.

Here’s the RA traces from Wireless controller when client is trying to connect:
Client entering into L2 authentication state and EAP process is started

```
2023/04/12 16:51:27.606414 {wncd_x_R0-0}{1}: [dot1x] [19224]: (info):
[Client_MAC:capwap_90000004] Entering request state
2023/04/12 16:51:27.606425 {wncd_x_R0-0}{1}: [dot1x] [19224]: (info):
[0000.0000.0000:capwap_90000004] Sending out EAPOL packet
2023/04/12 16:51:27.606494 {wncd_x_R0-0}{1}: [dot1x] [19224]: (info):
[Client_MAC:capwap_90000004] Sent EAPOL packet - Version : 3,EAPOL Type
: EAP, Payload Length : 1008, EAP-Type = EAP-TLS
2023/04/12 16:51:27.606496 {wncd_x_R0-0}{1}: [dot1x] [19224]: (info):
[Client_MAC:capwap_90000004] EAP Packet - REQUEST, ID : 0x25
2023/04/12 16:51:27.606536 {wncd_x_R0-0}{1}: [dot1x] [19224]: (info):
[Client_MAC:capwap_90000004] EAPOL packet sent to client
2023/04/12 16:51:27.640768 {wncd_x_R0-0}{1}: [dot1x] [19224]: (info):
[Client_MAC:capwap_90000004] Received EAPOL packet - Version : 1,EAPOL
Type : EAP, Payload Length : 6, EAP-Type = EAP-TLS
2023/04/12 16:51:27.640781 {wncd_x_R0-0}{1}: [dot1x] [19224]: (info):
[Client_MAC:capwap_90000004] EAP Packet - RESPONSE, ID : 0x25
```

When the Wireless controller sends the access request to the AAA server and the packet size is below 1500 bytes (which is the default MTU on the Wireless controller), the access challenge is received without any complications.

```
2023/04/12 16:51:27.641094 {wncd_x_R0-0}{1}: [radius] [19224]: (info):
RADIUS: Send Access-Request to 172.16.26.235:1812 id 0/6, len 552
2023/04/12 16:51:27.644693 {wncd_x_R0-0}{1}: [radius] [19224]: (info):
RADIUS: Received from id 1812/6 172.16.26.235:0, Access-Challenge, len
1141
```

Sometimes, a client may send its certificate for authentication. If the packet size exceeds the MTU, it will be fragmented before being sent further.

```
2023/04/12 16:51:27.758366 {wncd_x_R0-0}{1}: [radius] [19224]: (info):
RADIUS: Send Access-Request to 172.16.26.235:1812 id 0/8, len 2048
2023/04/12 16:51:37.761885 {wncd_x_R0-0}{1}: [radius] [19224]: (info):
RADIUS: Started 5 sec timeout
2023/04/12 16:51:42.762096 {wncd_x_R0-0}{1}: [radius] [19224]: (info):
RADIUS: Retransmit to (172.16.26.235:1812,1813) for id 0/8
2023/04/12 16:51:32.759255 {wncd_x_R0-0}{1}: [radius] [19224]: (info):
RADIUS: Retransmit to (172.16.26.235:1812,1813) for id 0/8
2023/04/12 16:51:32.760328 {wncd_x_R0-0}{1}: [radius] [19224]: (info):
RADIUS: Started 5 sec timeout
2023/04/12 16:51:37.760552 {wncd_x_R0-0}{1}: [radius] [19224]: (info):
RADIUS: Retransmit to (172.16.26.235:1812,1813) for id 0/8
2023/04/12 16:51:42.762096 {wncd_x_R0-0}{1}: [radius] [19224]: (info):
RADIUS: Retransmit to (172.16.26.235:1812,1813) for id 0/8
```

We have noticed that the packet size is 2048, which surpasses the default MTU. Consequently, there has been no response from the AAA server. The Wireless controller will persistently resend the access request until it reaches the maximum number of retries. Due to the absence of a response, the Wireless controller will ultimately reset the EAPOL process.

```
2023/04/12 16:51:45.762890 {wncd_x_R0-0}{1}: [dot1x] [19224]: (info):
[Client_MAC:capwap_90000004] Posting EAPOL_START on Client
2023/04/12 16:51:45.762956 {wncd_x_R0-0}{1}: [dot1x] [19224]: (info):
[Client_MAC:capwap_90000004] Entering init state
2023/04/12 16:51:45.762965 {wncd_x_R0-0}{1}: [dot1x] [19224]: (info):
[Client_MAC:capwap_90000004] Posting !AUTH_ABORT on Client
2023/04/12 16:51:45.762969 {wncd_x_R0-0}{1}: [dot1x] [19224]: (info):
[Client_MAC:capwap_90000004] Entering restart state
```

This process goes in loop and client is stuck in authentication phase only.

The Embedded Packet Capture captured on the Wireless controller shows that after several access requests and challenge exchanges with an MTU less than 1500 bytes, the Wireless controller sends an access request exceeding 1500 bytes, which contains the client's certificate. This larger packet undergoes fragmentation. However, there is no response to this particular access request. The Wireless controller continues to resend this request until it reaches the maximum number of retries, after which the EAP-TLS session restarts. This sequence of events keeps repeating, indicating that there is an EAP-TLS loop occurring as the client attempts to authenticate. Please refer to the concurrent packet captures from both the Wireless controller and ISE provided below for a clearer understanding.

Wireless controller EPC:

radius.code == 1

No.	Time	Protocol	Length	Info
109	12:21:27.510959	RADIUS	594	Access-Request id=3
110	12:21:27.510959	RADIUS	594	Access-Request id=3, Duplicate Request
117	12:21:27.554963	RADIUS	594	Access-Request id=4
118	12:21:27.554963	RADIUS	594	Access-Request id=4, Duplicate Request
125	12:21:27.599959	RADIUS	594	Access-Request id=5
126	12:21:27.599959	RADIUS	594	Access-Request id=5, Duplicate Request
135	12:21:27.640958	RADIUS	594	Access-Request id=6
136	12:21:27.640958	RADIUS	594	Access-Request id=6, Duplicate Request
143	12:21:27.676951	RADIUS	594	Access-Request id=7
144	12:21:27.676951	RADIUS	594	Access-Request id=7, Duplicate Request
154	12:21:27.758948	RADIUS	714	Access-Request id=8
796	12:21:32.759955	RADIUS	714	Access-Request id=8, Duplicate Request
1130	12:21:37.761954	RADIUS	714	Access-Request id=8, Duplicate Request
1868	12:21:42.762945	RADIUS	714	Access-Request id=8, Duplicate Request
2132	12:21:45.796955	RADIUS	538	Access-Request id=9
2133	12:21:45.796955	RADIUS	538	Access-Request id=9, Duplicate Request
2144	12:21:45.854951	RADIUS	760	Access-Request id=10
2145	12:21:45.854951	RADIUS	760	Access-Request id=10, Duplicate Request
2168	12:21:45.914945	RADIUS	594	Access-Request id=11
2169	12:21:45.914945	RADIUS	594	Access-Request id=11, Duplicate Request
2176	12:21:45.959941	RADIUS	594	Access-Request id=12

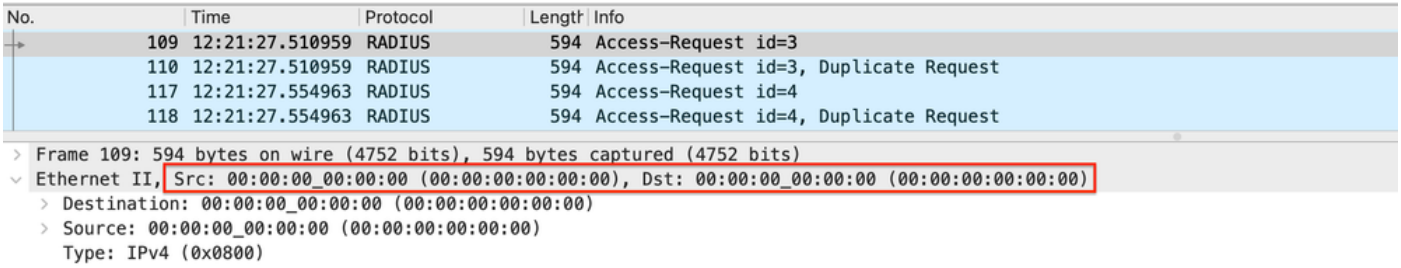
Packet Capture on WLC

We observe that the Wireless controller is sending several duplicate requests for a particular Access-request ID = 8

Note: On the EPC, we also notice that there is a single duplicate request for other IDs. This prompts the question: Is such duplication expected? The answer to whether this duplication is expected is yes, it is. The reason is that the capture was taken from the Wireless controller's GUI

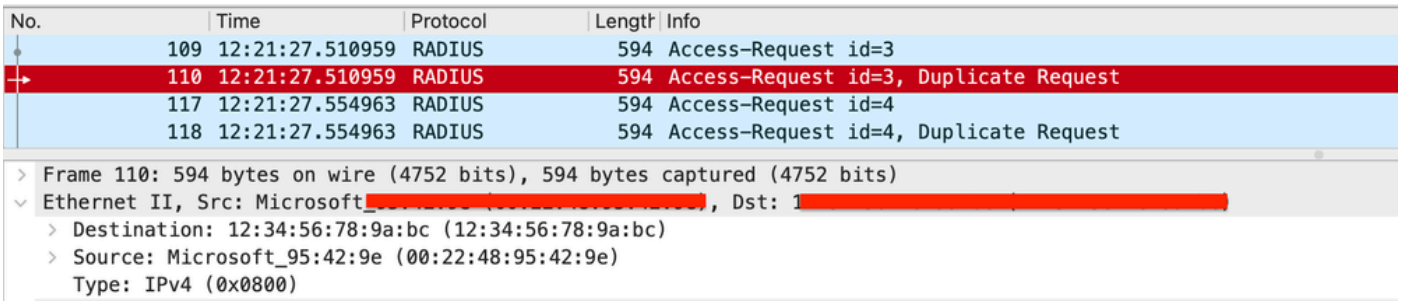
---

with the 'Monitor Control Plane' option selected. As a result, it is normal to observe several instances of RADIUS packets since they are being directed to the CPU. In such cases, Access requests must be seen with both source and destination MAC addresses set to 00:00:00.



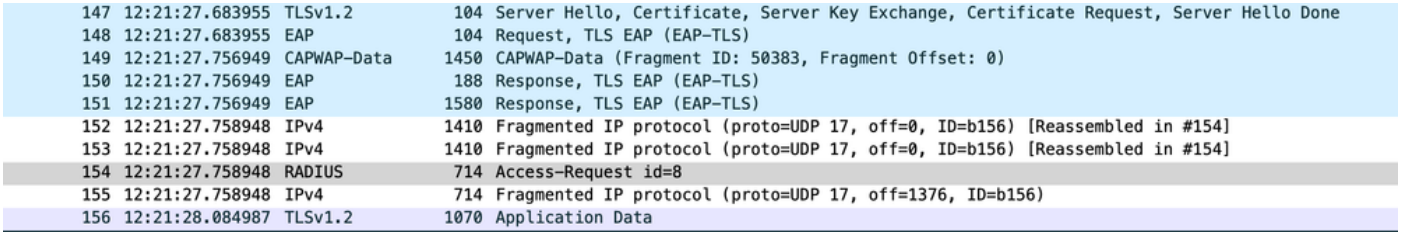
Radius Access-Request Punted to CPU on WLC

Only the Access requests with the specified source and destination MAC addresses must actually be sent out of the Wireless controller.



Radius Access-Request Sent to AAA Server

The Access requests in question, identified by ID = 8, which are sent out multiple times and for which no response was seen from AAA server. Upon further investigation, we observed that for Access-request ID=8, UDP fragmentation is occurring due to the size surpassing the MTU, as illustrated below:



Fragmentation taking Place on WLC Packet Capture

```
> Frame 152: 1410 bytes on wire (11280 bits), 1410 bytes captured (11280 bits)
v Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
  > Destination: 00:00:00_00:00:00 (00:00:00:00:00:00)
  > Source: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Type: IPv4 (0x0800)
v Internet Protocol Version 4, Src: 10.100.9.15, Dst: 172.16.26.235
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1396
    Identification: 0xb156 (45398)
  > 001. .... = Flags: 0x1, More fragments
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: UDP (17)
    Header Checksum: 0xc9b4 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.100.9.15
    Destination Address: 172.16.26.235
    [Reassembled IPv4 in frame: 154]
> Data (1376 bytes)
```

Fragmented Packet - I

```
> Frame 153: 1410 bytes on wire (11280 bits), 1410 bytes captured (11280 bits)
v Ethernet II, Src: Microsoft_[redacted] Dst: [redacted]
  > Destination: 12:34:56:78:9a:bc (12:34:56:78:9a:bc)
  > Source: Microsoft_95:42:9e (00:22:48:95:42:9e)
    Type: IPv4 (0x0800)
v Internet Protocol Version 4, Src: 10.100.9.15, Dst: 172.16.26.235
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1396
    Identification: 0xb156 (45398)
  > 001. .... = Flags: 0x1, More fragments
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: UDP (17)
    Header Checksum: 0xc9b4 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.100.9.15
    Destination Address: 172.16.26.235
    [Reassembled IPv4 in frame: 154]
```

Fragmented Packet - II

```
    152 12:21:27.758948 IPv4            1410 Fragmented IP protocol (proto=UDP 17, off=0, ID=b156) [Reassembled in #154]
    153 12:21:27.758948 IPv4            1410 Fragmented IP protocol (proto=UDP 17, off=0, ID=b156) [Reassembled in #154]
    154 12:21:27.758948 RADIUS           714 Access-Request id=8
    155 12:21:27.758948 IPv4            714 Fragmented IP protocol (proto=UDP 17, off=1376, ID=b156)

> Frame 154: 714 bytes on wire (5712 bits), 714 bytes captured (5712 bits)
> Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
v Internet Protocol Version 4, Src: 10.100.9.15, Dst: 172.16.26.235
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 700
    Identification: 0xb156 (45398)
  > 000. .... = Flags: 0x0
    ...0 0000 1010 1100 = Fragment Offset: 1376
    Time to Live: 64
    Protocol: UDP (17)
    Header Checksum: 0xebc0 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.100.9.15
    Destination Address: 172.16.26.235
v [3 IPv4 Fragments (2056 bytes): #152(1376), #153(1376), #154(680)]
    [Frame: 152, payload: 0-1375 (1376 bytes)]
  > [Frame: 153, payload: 0-1375 (1376 bytes)]
    [Frame: 154, payload: 1376-2055 (680 bytes)]
    [Fragment count: 3]
    [Reassembled IPv4 length: 2056]
```

Reassembled Packet

To cross verify, we reviewed the ISE logs and discovered that the access request, which had been fragmented on the Wireless controller, was not being received by the ISE at all.

ISE TCP Dumps

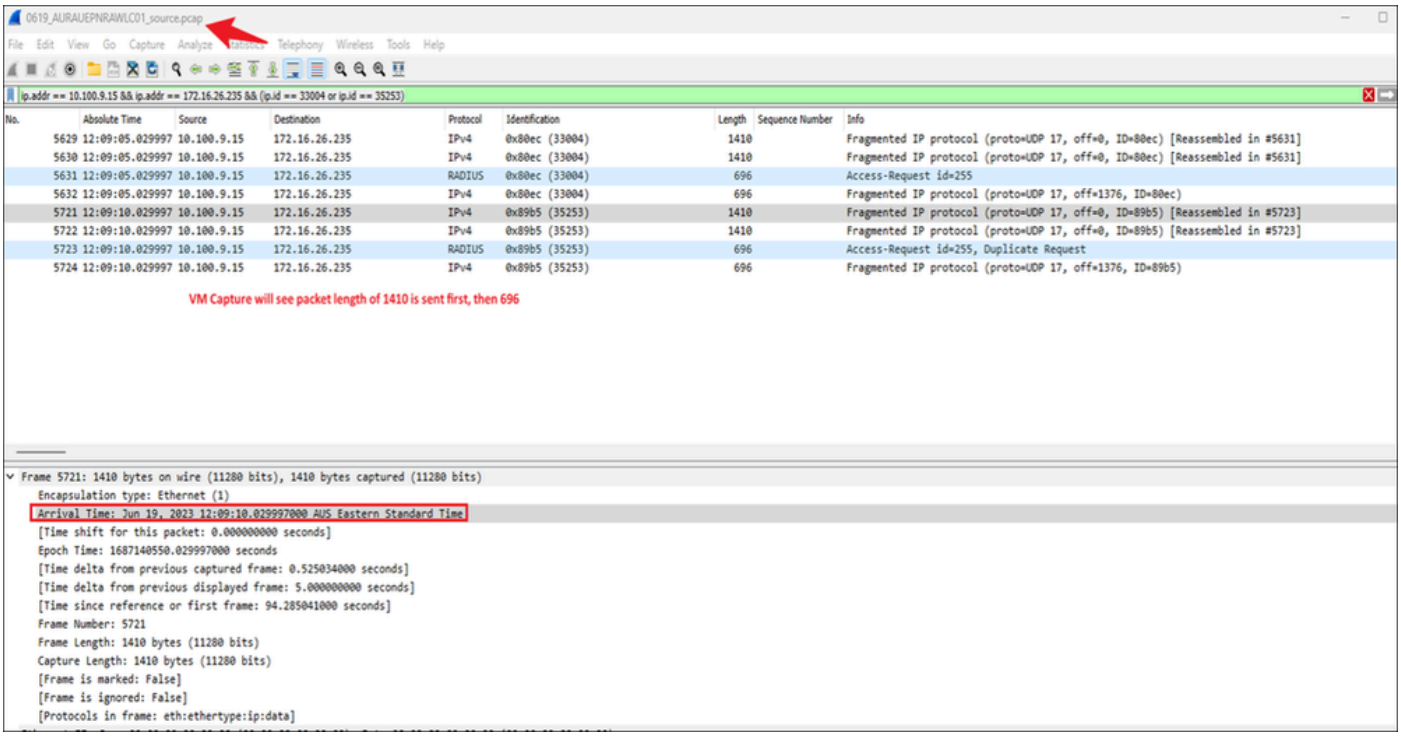
radius.code == 1

o.	Time	Protocol	Length	Info
1	12:21:27.387158	RADIUS	538	Access-Request id=0
3	12:21:27.428304	RADIUS	760	Access-Request id=1
5	12:21:27.492019	RADIUS	594	Access-Request id=2
7	12:21:27.527949	RADIUS	594	Access-Request id=3
9	12:21:27.572272	RADIUS	594	Access-Request id=4
11	12:21:27.617147	RADIUS	594	Access-Request id=5
13	12:21:27.657917	RADIUS	594	Access-Request id=6
15	12:21:27.694381	RADIUS	594	Access-Request id=7
17	12:21:45.814195	RADIUS	538	Access-Request id=9
19	12:21:45.871163	RADIUS	760	Access-Request id=10
21	12:21:45.932076	RADIUS	594	Access-Request id=11
23	12:21:45.977012	RADIUS	594	Access-Request id=12
25	12:21:46.018562	RADIUS	594	Access-Request id=13

Captures on ISE End

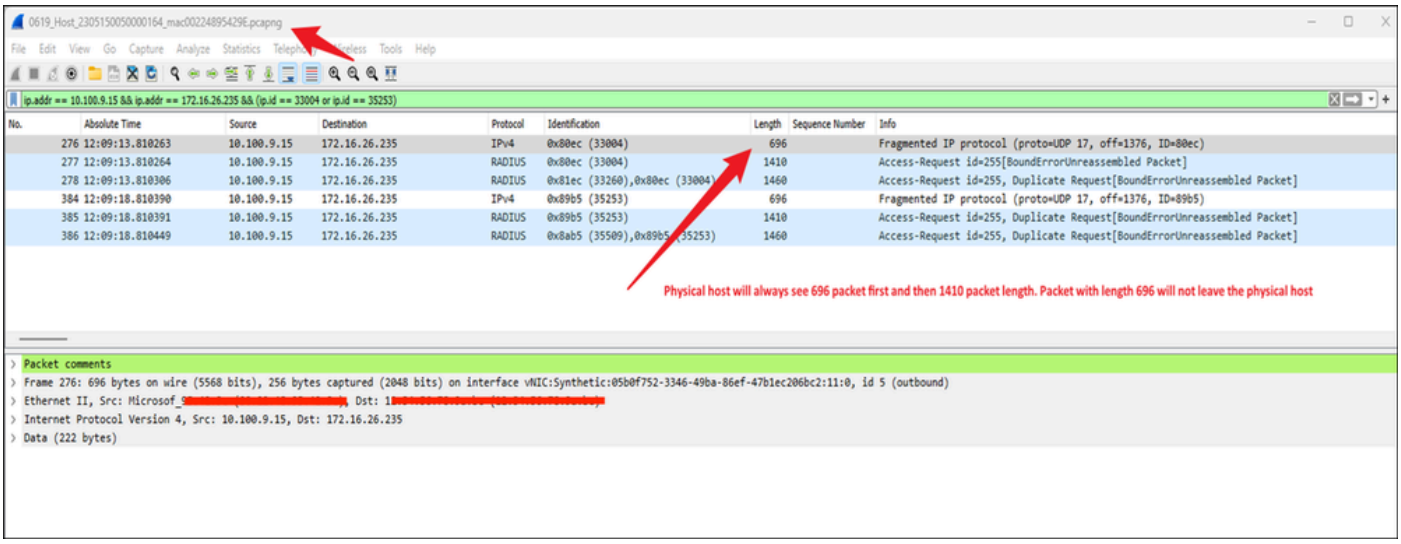
Azure Side Capture with analysis:

The Azure team conducted a capture on the physical host within Azure. The data captured on the vSwitch within the Azure host indicates that the UDP packets are arriving out of sequence. Because these UDP fragments are not in the correct order, Azure is discarding them. Below are the captures from both the Azure end and the Wireless controller, taken simultaneously for access request ID = 255, where the issue of packets being out of order is clearly evident:
The Encapsulated Packet Capture (EPC) on the Wireless controller displays the sequence in which the fragmented packets are leaving from the Wireless controller.



Sequence of Fragmented Packets on WLC

On the physical host, the packets are not arriving in the proper sequence



Captures on Azure End

Since the packets are arriving in the wrong order, and the physical node is programmed to reject any out-of-order frames, the packets gets dropped immediately. This disruption causes the authentication process to fail, leaving the client unable to progress beyond the authentication phase.

Workaround suggested from Wireless controller end:

Starting with version 17.11.1, we are implementing support for Jumbo Frames in Radius/AAA packets. This feature allows the c9800 controller to avoid fragmenting AAA packets, provided that the following configuration is set on the controller. Please note that to avoid fragmentation of these packets entirely, it is essential to ensure that every network hop, including the AAA server, is compatible with Jumbo Frame packets. For ISE, Jumbo Frame support begins with version 3.1 onwards.

Interface configuration on Wireless controller:

C9800-CL(config)#interface <Interface Name> C9800-CL(config-if) # mtu <bytes> C9800-CL(config-if) # ip mtu <bytes> [1500 to 9000]

AAA server config on Wireless controller:

C9800-CL(config)# aaa group server radius <Radius Group Name> C9800-CL(config-sg-radius) # server name <Server Name> C9800-CL(config-sg-radius) # ip radius source-interface <Interface Name>

Here is a brief look at a Radius packet when the MTU (Maximum Transmission Unit) is configured to 3000 bytes on a Wireless LAN Controller (WLC). Packets smaller than 3000 bytes were sent seamlessly without the need for fragmentation:



1020	10:08:11.177984	RADIUS	2075	Access-Request id=199
1021	10:08:11.177984	RADIUS	2075	Access-Request id=199, Duplicate Request
1119	10:08:16.194981	RADIUS	2075	Access-Request id=199, Duplicate Request
1120	10:08:16.194981	RADIUS	2075	Access-Request id=199, Duplicate Request
1223	10:08:21.179983	RADIUS	2075	Access-Request id=199, Duplicate Request
1224	10:08:21.179983	RADIUS	2075	Access-Request id=199, Duplicate Request
1451	10:08:26.180990	RADIUS	2075	Access-Request id=199, Duplicate Request
1452	10:08:26.180990	RADIUS	2075	Access-Request id=199, Duplicate Request
2470	10:08:31.181982	RADIUS	2075	Access-Request id=199, Duplicate Request

Packet Capture on WLC with Increased MTU

By setting the configuration in this way, the Wireless controller transmits packets without fragmenting them, sending them intact. **However, because Azure cloud does not support jumbo frames, this solution cannot be implemented.**

Solution:

- From the Wireless controller's Encapsulated Packet Capture (EPC), we observed that the packets are being sent in the correct order. It is then the responsibility of the receiving host to reassemble them properly and continue with processing, which, in this case, is not occurring on the Azure side.
- To address the issue of out-of-order UDP packets, the enable-udp-fragment-reordering option needs to be activated on Azure.
- You must reach out to Azure support team for assistance with this matter. Microsoft has acknowledged this problem.



Note: It must be noted that this problem is not exclusive to the interoperability with Cisco 9800 Wireless LAN Controller (WLC). Similar issues with out-of-order UDP packets have been encountered on different radius servers, including ISE, Forti Authenticator, and RTSP servers, particularly when they operate within the Azure environment.
