解决分段问题：影响使用Azure的c9800无线控制器

目录

简介

症状

ISE服务器出错

详细日志分析：

- 无线控制器EPC:
- ISE TCP转储
- 带分析的Azure Side Capture:

无线控制器端建议的解决方法：

解决方案：

简介

本文档介绍Azure平台的一个已知问题，该问题导致由于处理无序分段错误而导致数据包丢失。

症状

受影响的产品:托管在Azure上的Catalyst 9800-CL无线控制器或托管在Azure上的身份服务引擎。

SSID设置：配置为使用中央身份验证的802.1x EAP-TLS。

执行：使用托管在Azure平台上的9800-CL以及基于EAP-TLS的SSID时，可能会遇到连接问题。客户端在身份验证阶段可能会遇到困难。

ISE服务器出错

错误代码5411，指示请求方在EAP-TLS证书交换期间已停止与ISE的通信。

详细日志分析：

下面是其中一个受影响配置的图示：在9800无线控制器中，为802.1x设置SSID，为EAP-TLS配置AAA服务器。当客户端尝试进行身份验证时，特别是在证书交换阶段，客户端发送超过无线控制器上最大传输单位(MTU)大小的证书。9800无线控制器随后会对此大型数据包进行分段，然后按顺序将分段发送到AAA服务器。但是，这些分段在物理主机上的到达顺序不正确，导致丢包。

以下是客户端尝试连接时无线控制器的RA跟踪：

进入L2身份验证状态和EAP进程的客户端已启动

```
2023/04/12 16:51:27.606414 {wncd_x_R0-0}{1}:[dot1x] [19224]:(信息
)[Client_MAC:capwap_90000004]进入请求状态
2023/04/12 16:51:27.606425 {wncd_x_R0-0}{1}:[dot1x] [19224]:(信息
)[0000.0000.0000:capwap_90000004]发送EAPOL数据包
2023/04/12 16:51:27.606494 {wncd_x_R0-0}{1}:[dot1x] [19224]:(信息
)[Client_MAC:capwap_90000004]发送的EAPOL数据包 — 版本：3,EAPOL类型：EAP，负载
长度：1008,EAP-Type = EAP-TLS
2023/04/12 16:51:27.606496 {wncd_x_R0-0}{1}:[dot1x] [19224]:(信息
)[Client_MAC:capwap_90000004] EAP数据包 — 请求，ID:0x25
2023/04/12 16:51:27.606536 {wncd_x_R0-0}{1}:[dot1x] [19224]:(信息
)[Client_MAC:capwap_90000004]发送到客户端的EAPOL数据包
2023/04/12 16:51:27.640768 {wncd_x_R0-0}{1}:[dot1x] [19224]:(信息
)[Client_MAC:capwap_90000004]已收到EAPOL数据包 — 版本：1,EAPOL类型：EAP，负载
长度：6,EAP-Type = EAP-TLS
2023/04/12 16:51:27.640781 {wncd_x_R0-0}{1}:[dot1x] [19224]:(信息
)[Client_MAC:capwap_90000004] EAP数据包 — 响应，ID:0x25
```

当无线控制器向AAA服务器发送访问请求且数据包大小低于1500字节（无线控制器上的默认MTU）时，接收访问质询时不会出现任何问题。

```
2023/04/12 16:51:27.641094 {wncd_x_R0-0}{1}:[radius] [19224]:(信息
)RADIUS:将访问请求发送到172.16.26.235:1812 id 0/6,len 552
2023/04/12 16:51:27.644693 {wncd_x_R0-0}{1}:[radius] [19224]:(信息
)RADIUS:接收自id 1812/6 172.16.26.235:0,Access-Challenge，len 1141
```

有时，客户端可能会发送其证书以进行身份验证。如果数据包大小超过MTU，则数据包在进一步发送之前会先进行分段。

```
2023/04/12 16:51:27.758366 {wncd_x_R0-0}{1}:[radius] [19224]:(信息
)RADIUS:发送访问请求到172.16.26.235:1812 id 0/8,len 2048
2023/04/12 16:51:37.761885 {wncd_x_R0-0}{1}:[radius] [19224]:(信息
)RADIUS:已启动5秒超时
2023/04/12 16:51:42.762096 {wncd_x_R0-0}{1}:[radius] [19224]:(信息
)RADIUS:ID 0/8重新传输到(172.16.26.235:1812,1813)
2023/04/12 16:51:32.759255 {wncd_x_R0-0}{1}:[radius] [19224]:(信息
)RADIUS:ID 0/8重新传输到(172.16.26.235:1812,1813)
2023/04/12 16:51:32.760328 {wncd_x_R0-0}{1}:[radius] [19224]:(信息
)RADIUS:已启动5秒超时
2023/04/12 16:51:37.760552 {wncd_x_R0-0}{1}:[radius] [19224]:(信息
)RADIUS:ID 0/8重新传输到(172.16.26.235:1812,1813)
2023/04/12 16:51:42.762096 {wncd_x_R0-0}{1}:[radius] [19224]:(信息
)RADIUS:ID 0/8重新传输到(172.16.26.235:1812,1813)
```

我们注意到数据包大小为2048，超过默认MTU。因此，没有来自AAA服务器的响应。无线控制器将持续重新发送访问请求，直到达到最大重试次数。由于没有响应，无线控制器将最终重置EAPOL进程。

```
2023/04/12 16:51:45.762890 {wncd_x_R0-0}{1}:[dot1x] [19224]:(信息)[Client_MAC:capwap_90000004]在客户端上发布EAPOL_START
2023/04/12 16:51:45.762956 {wncd_x_R0-0}{1}:[dot1x] [19224]:(信息)[Client_MAC:capwap_90000004]正在进入init状态
2023/04/12 16:51:45.762965 {wncd_x_R0-0}{1}:[dot1x] [19224]:(信息)[Client_MAC:capwap_90000004]在客户端上发布！AUTH_ABORT
2023/04/12 16:51:45.762969 {wncd_x_R0-0}{1}:[dot1x] [19224]:(信息)[Client_MAC:capwap_90000004]正在进入重新启动状态
```

此过程处于循环状态，客户端仅停滞在身份验证阶段。

在无线控制器上捕获的嵌入式数据包捕获显示，在与小于1500字节的MTU进行多次访问请求和质询交换后，无线控制器发送超过1500字节的访问请求，其中包含客户端证书。此较大的数据包将进行分段。但是，此特定访问请求没有响应。无线控制器继续重新发送此请求，直到达到最大重试次数，然后EAP-TLS会话重新启动。此事件序列不断重复，表明客户端尝试进行身份验证时发生了EAP-TLS环路。请参阅以下提供的无线控制器和ISE的并发数据包捕获，了解更多详情。

无线控制器EPC:

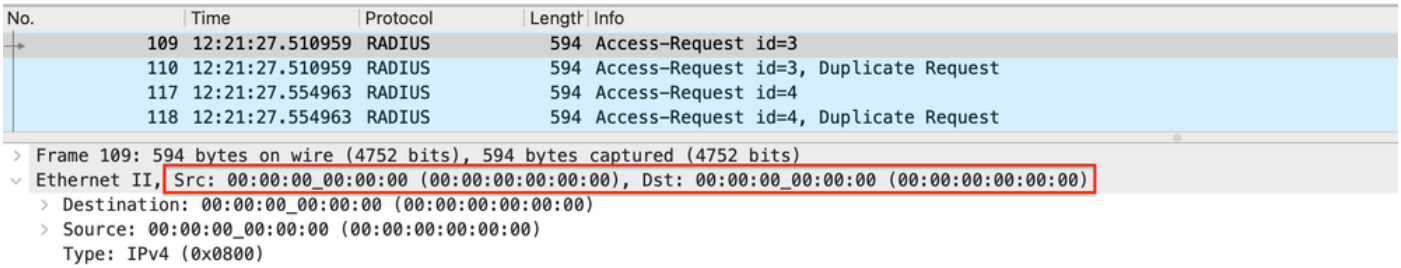
radius.code == 1

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

WLC上的数据包捕获

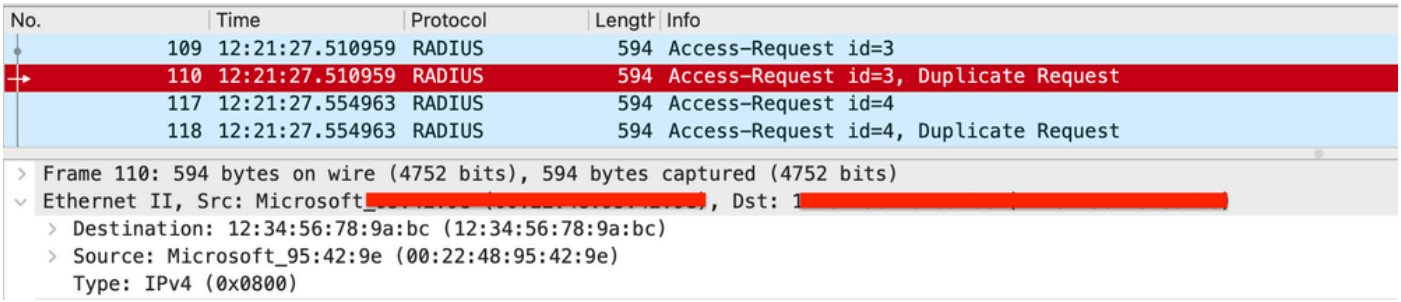
我们观察到无线控制器正在发送对特定访问请求ID = 8的多个重复请求

注意：在EPC上，我们还注意到对其他ID存在单个重复请求。这就引出了一个问题：是否期望出现这种重复？至于是否应该出现这种重复，答案是肯定的。原因是捕获是从无线控制器的GUI获取的，并且选中了“Monitor Control Plane”（监控控制平面）选项。因此，观察RADIUS数据包的几个实例是正常的，因为它们被定向到CPU。在这种情况下，必须看到源和目标MAC地址都设置为00:00:00的访问请求。



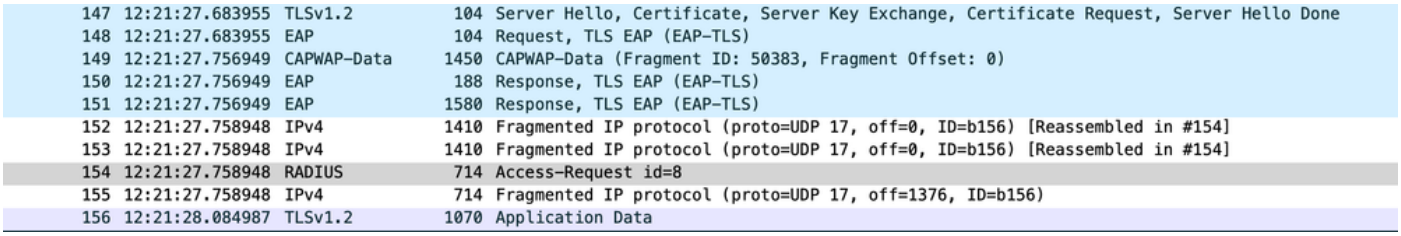
Radius访问请求传送到WLC上的CPU

只有具有指定源和目标MAC地址的访问请求才能实际从无线控制器发出。



发送到AAA服务器的Radius访问请求

有问题的访问请求（由ID = 8标识）被多次发送并且未从AAA服务器看到响应。进一步调查后，我们发现对于Access-request ID=8,UDP分段是由大小超出MTU造成的，如下所示：



WLC数据包捕获发生分段

```
> 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)
```

分段数据包 — 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]
```

分段数据包 — 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]
```

重组数据包

为了交叉验证，我们检查了ISE日志，发现无线控制器上分段的访问请求根本未被ISE接收。

ISE TCP转储

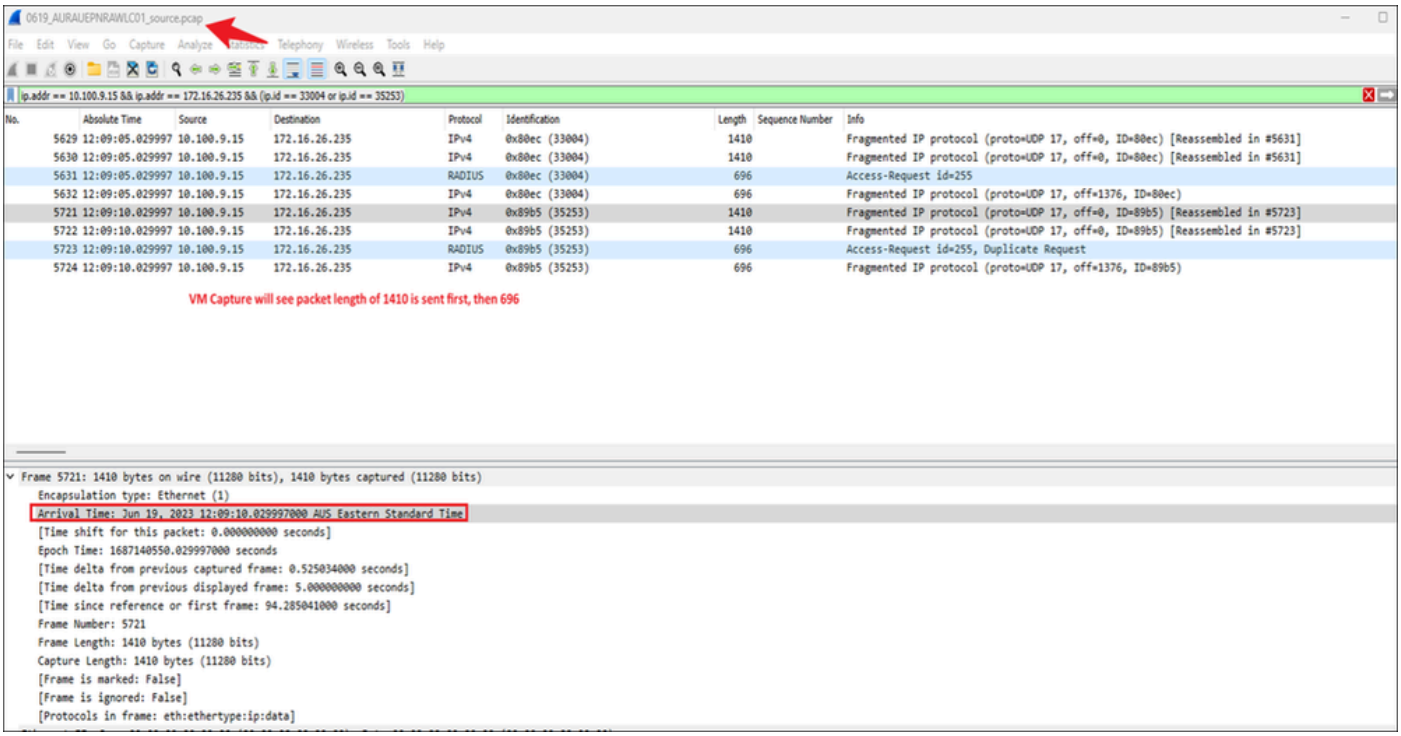
radius.code == 1

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

在ISE端捕获

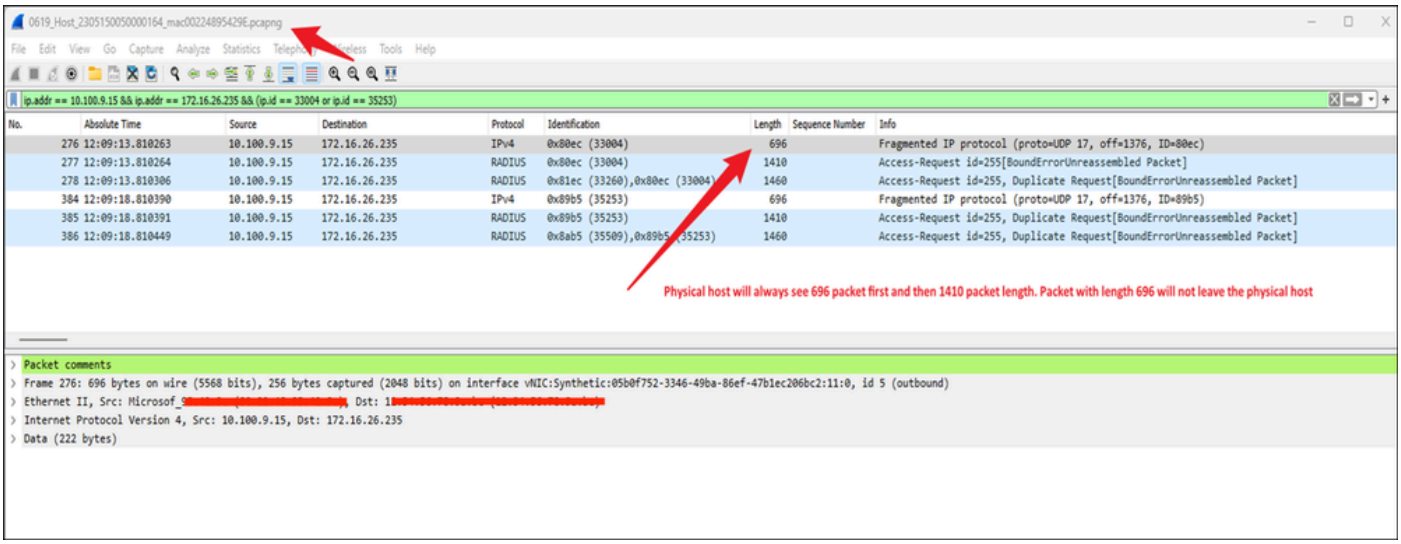
带分析的Azure Side Capture:

Azure团队在Azure内的物理主机上执行了捕获。在Azure主机内的vSwitch上捕获的数据表明UDP数据包的到达顺序混乱。由于这些UDP分段的顺序不正确，Azure正在丢弃它们。以下是Azure端和无线控制器同时捕获的访问请求ID = 255，其中明显存在数据包顺序混乱的问题：
无线控制器上的封装数据包捕获(EPC)显示分段数据包从无线控制器离开的顺序。



WLC上的分段数据包序列

在物理主机上，数据包的到达顺序不正确



Azure End上的捕获

由于数据包的到达顺序错误，并且物理节点被编程为拒绝任何无序帧，因此数据包会立即被丢弃。这种中断会导致身份验证过程失败，使客户端无法继续身份验证阶段。

无线控制器端建议的解决方法：

从版本17.11.1开始，我们将实施对Radius/AAA数据包中的巨帧的支持。此功能允许c9800控制器避免对AAA数据包进行分段，前提是在控制器上设置以下配置。请注意，要完全避免这些数据包的分段，必须确保每个网络跃点（包括AAA服务器）与巨型帧数据包兼容。对于ISE，从3.1版本开始支持巨型帧。
无线控制器上的接口配置：

C9800-CL(config)#interface

C9800-CL(config-if) # mtu

C9800-CL(config-if) # ip mtu

[1500 to 9000]

无线控制器上的AAA服务器配置：

C9800-CL(config)# aaa group server radius

C9800-CL(config-sg-radius) # server name

C9800-CL(config-sg-radius) # ip radius source-interface

下面简要介绍当无线局域网控制器(WLC)上的MTU（最大传输单位）配置为3000字节时的Radius数据包。 小于3000字节的数据包无需分段即可无缝发送：

```
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
```

WLC上的数据包捕获（增加MTU）

通过以这种方式设置配置，无线控制器可以传输数据包，而不会对其进行分段，并完整地发送这些数据包。但是，由于Azure云不支持巨型帧，因此无法实施此解决方案。

解决方案：

- 通过无线控制器的封装数据包捕获(EPC)，我们观察到数据包按正确顺序发送。然后，接收主机有责任正确重组它们，并继续处理，在这种情况下，Azure端不会发生此情况。
- 要解决顺序混乱的UDP数据包问题enable-udp-fragment-reordering，需要在Azure上激活该选项。
- 您必须联系Azure支持团队以获取有关此问题的帮助。Microsoft已承认此问题。



注意：必须注意的是，此问题并不只限于与Cisco 9800无线局域网控制器(WLC)的互操作性。 不同RADIUS服务器（包括ISE、Forti Authenticator和RTSP服务器）上也遇到类似顺序混乱的UDP数据包问题，尤其是当这些服务器在Azure环境中运行时。

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。