9800 WLC의 RADIUS MTU 및 단편화 이해

목차

소개

이 문서에서는 WLC가 RADIUS 서버로 전송하는 RADIUS 패킷의 MTU를 구성하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

Cisco에서는 다음 항목에 대해 기본적으로 이해하고 있는 것이 좋습니다.

- 9800 WLC(Wireless LAN Controller) AAA 컨피그레이션
- AAA(Authentication, Authorization and Accounting) RADIUS 개념
- 확장 가능 한 인증 프로토콜 EAP
- 최대 전송 단위(MTU)

사용되는 구성 요소

- Cisco ISE(Identity Service Engineer) 3.2
- Catalyst 9800 Wireless Controller Series(Catalyst 9800-L)
- Cisco IOS® XE 17.15.2
- Windows 11 무선 클라이언트

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

배경

이 문서에서는 Cisco ISE에서 사용한 RADIUS(Remote Authentication Dial-In User Service) 서버를 사용합니다. 먼저 EAP(Extensible Authentication Protocol) 프로세스 중에 외부 개입 없이 패킷이 어떻게 흐르는지 설명합니다. 다음은 WLC가 RADIUS 서버에 보내는 액세스 요청의 크기를 변경하는 컨피그레이션 옵션입니다. 이 옵션은 IOS-XE 버전 17.11에 추가되었습니다.

9800 RADIUS MTU

일반적으로 RADIUS 패킷의 MTU는 작지만 MTU에 도달하지 않으므로 중요하지 않습니다. 그러나 한 쪽에서 인증서를 전송해야 하는 경우(일반적으로 2-5KB), 디바이스는 해당 인증서를 프래그먼트화하여 MTU에 맞게 받아야 합니다.

EAP-TLS(EAP Transport Layer Security)의 경우와 같이 클라이언트가 RADIUS 서버에 인증서를 전송해야 할 때 WLC는 패킷을 보내야 하는 RADIUS 데이터의 양으로 인해 패킷을 다시 프래그먼트화해야 하는 상황을 나타냅니다. 17.11까지는 네트워크 관리자가 이 프로세스를 거의 제어할 수 없었지만, 이제 엔지니어에게 WLC에서 보낸 액세스 요청의 크기를 조작할 수 있는 옵션이 주어졌습니다.

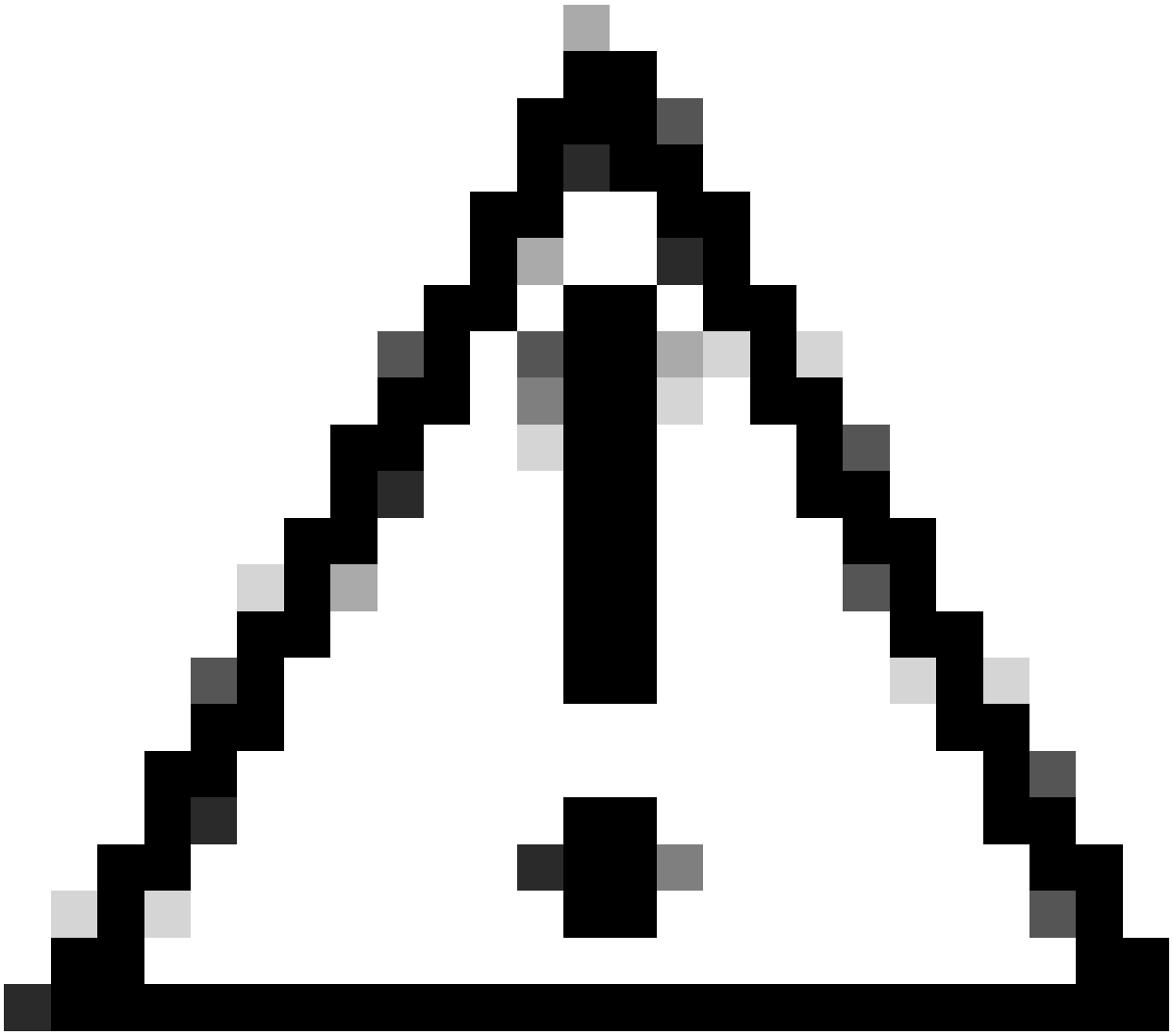
EAP-TLS 패킷 흐름

이는 무선 인프라에서 패킷이 어떻게 나타나는지, 어떻게 처리되는지를 심층적으로 살펴보는 것입니다. 이 문서에 소개된 변경 사항을 제대로 이해하려면 dot1x, 특히 EAP-TLS를 사용할 때 무선 인증 프로세스 중에 패킷 흐름을 파악하는 것이 중요합니다.

Cisco 무선 인프라에서 EAP 및 RADIUS 패킷 흐름이 어떻게 작동하는지 이미 충분히 알고 있는 경우 동작 변경 섹션으로 이동하여 17.11에 추가된 사항을 설명함으로써 네트워크 관리자에게 RADIUS MTU를 더 효과적으로 제어할 수 있도록 합니다. 먼저 EAP ID(EAP ID)를 확인합니다.

EAP-ID

EAP-ID는 인증자(이 경우 WLC)에 의해 시작됩니다. 이는 EAP 프로세스의 첫 번째 부분이어야 합니다. 무선 클라이언트가 EAPOL-Start를 전송하는 경우도 있습니다. 이는 일반적으로 클라이언트가 EAP-ID 요청을 받지 못했거나 처음부터 다시 시작하려고 함을 의미합니다.



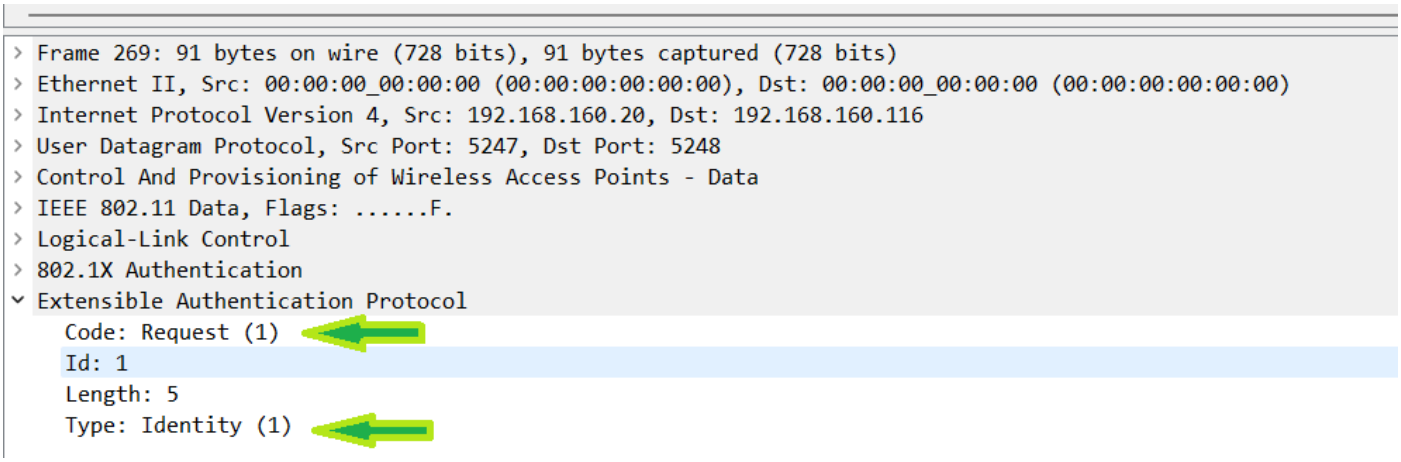
주의: EAP-ID 패킷과 EAP 패킷 ID 간에는 차이가 있습니다. EAP-ID 패킷은 EAP 패킷 ID가 네트워크를 통과할 때 특정 패킷을 추적하는 데 사용되는 번호인 신청자를 식별하는 데 사용됩니다.

EAP-ID 요청

먼저, 무선 클라이언트 장치는 정상적인 연계 프로세스를 이용하여 네트워크에 접속한다. WLAN(Wireless Local Area Network)이 dot1x에 대해 구성된 경우 WLC는 먼저 클라이언트가 누구인지 알아야 RADIUS 서버에서 액세스를 요청할 수 있습니다. 이 정보를 찾기 위해 WLC는 클라이언트 및 EAP-ID 요청을 보냅니다.

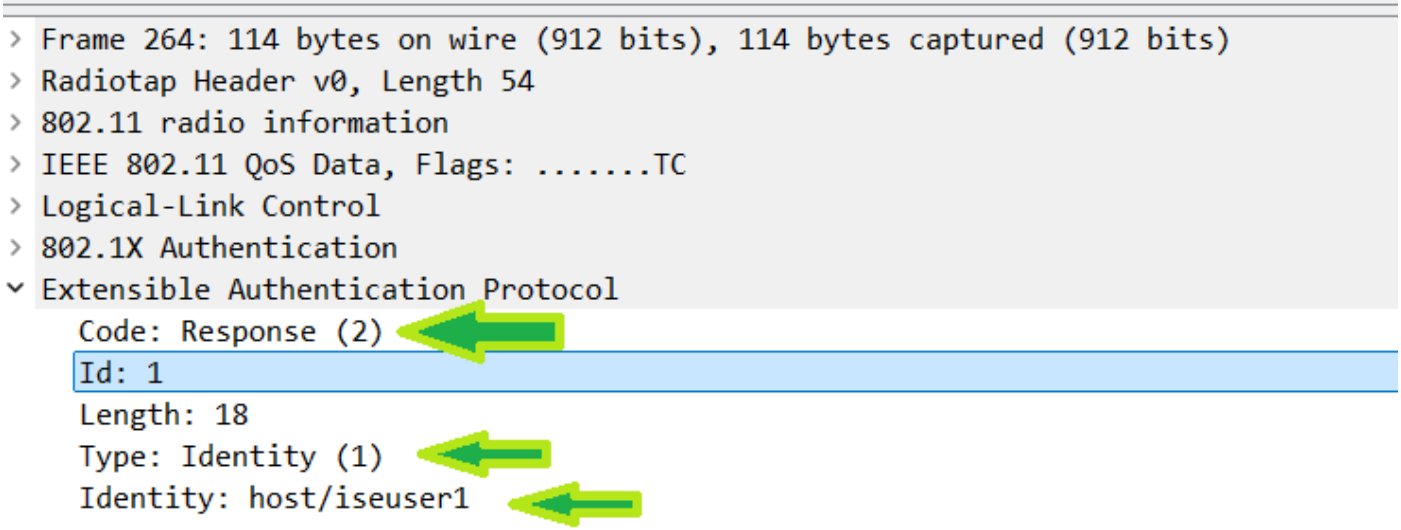
클라이언트는 EAP-ID 응답으로 응답해야 합니다. 그러면 WLC에서 액세스 요청을 작성하여 ISE에 전송하는 데 필요한 기능이 제공됩니다. EAP-ID 요청은 정상적인 PEAP 인증에서 클라이언트가 사용자 이름과 비밀번호를 입력하도록 요청되는 경우입니다.

그러나 이 논의는 EAP-TLS에 대한 것이므로 여기에 있는 EAP-ID 응답은 사용자 ID만 가질 수 있습니다. 데모에서 사용자 ID는 iseuser1입니다. 이 패킷에서는 WLC가 무선 클라이언트에 누구인지를 묻는 EAP-ID 요청을 볼 수 있습니다. 이 클라이언트는 무선 클라이언트이므로 WLC는 CAPWAP에서 요청을 캡슐화하고 무선 전송을 위해 액세스 포인트(AP)에 전송합니다. EAP 데이터에서 코드 1은 요청임을 나타내고 유형 1은 ID를 나타냅니다.



EAP-ID 응답

다음으로, 무선 클라이언트가 EAP-ID 응답으로 응답하는 것을 볼 수 있습니다. EAP 데이터에서 코드가 2로 변경되어 응답임을 나타내지만 유형은 1로 유지되며 ID에 대한 것임을 나타냅니다. 여기서는 클라이언트가 사용 중인 사용자 이름까지 볼 수 있습니다. 이러한 패킷에 대해 한 가지 더 확인할 사항은 EAP 패킷의 ID 번호입니다. EAP-ID 교환의 경우 항상 1이지만 나중에 ISE가 연결되면 이 숫자가 다른 값으로 변경됩니다.



두 패킷 모두 크기가 작으므로 MTU는 네트워크에서 사용되는 1500바이트 미만이므로 여기에 아무런 영향을 주지 않습니다.

액세스 요청 및 액세스 챌린지

클라이언트와의 통신은 EAP이며 WLC와 ISE 간의 통신은 RADIUS입니다. RADIUS 통신에는 access-request 및 access-challenge 패킷이 사용됩니다. WLC는 신청자로부터 EAP 패킷을 수신하

고 RADIUS 액세스 요청을 사용하여 ISE에 전달합니다. 작동 중인 네트워크에서 ISE는 액세스 챌린지로 응답합니다.

액세스 요청

이제 WLC는 클라이언트의 ID를 알고 있으므로 해당 클라이언트가 네트워크에서 허용되는지 RADIUS 서버에 문의해야 합니다. 이를 위해 WLC는 액세스 요청 패킷을 전송하여 해당 클라이언트에 대한 액세스를 요청합니다. WLC가 EAP 데이터와 함께 전송할 다른 데이터가 있습니다. 이를 통틀어 속성 값 쌍, AVP 또는 AV 쌍이라고 합니다.

이 문서는 이 논의의 범위를 벗어나는 것이므로 AVP로 멀리 이동하지는 않을 것입니다. 여기서 사용자 이름(EAP 데이터)이 포함되어 RADIUS 서버(이 경우에는 ISE)로 전송되는지 확인하기만 하면 됩니다. 또한 EAP-ID 번호 1도 ISE로 전송되는 것을 볼 수 있습니다. EAP 패킷 ID를 살펴보았을 때도 역시 1이었습니다. 여기서 마지막으로 주의해야 할 점은 WLC가 이 모든 AVP를 추가했기 때문에 클라이언트가 전송한 114바이트 패킷이 이제 ISE로 전송되기 전에 488바이트 패킷으로 변경된다는 것입니다.

```
> Frame 281: 506 bytes on wire (4048 bits), 506 bytes captured (4048 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)
> Internet Protocol Version 4, Src: 192.168.160.20, Dst: 192.168.160.88
> User Datagram Protocol, Src Port: 58038, Dst Port: 1812
v RADIUS Protocol
    Code: Access-Request (1)
    Packet identifier: 0x24 (36)
    Length: 464
    Authenticator: 48f74e792b11604d9188e4d947629485
    [The response to this request is in frame 285]
  v Attribute Value Pairs
    v AVP: t=User-Name(1) l=15 val=host/iseuser1
        Type: 1
        Length: 15
        User-Name: host/iseuser1
    > AVP: t=Service-Type(6) l=6 val=Framed(2)
    > AVP: t=Vendor-Specific(26) l=27 vnd=ciscoSystems(9)
    > AVP: t=Framed-MTU(12) l=6 val=576
    v AVP: t=EAP-Message(79) l=20 Last Segment[1]
        Type: 79
        Length: 20
        EAP fragment: 0201001201686f73742f6973657573657231
      v Extensible Authentication Protocol
          Code: Response (2)
          Id: 1
          Length: 18
          Type: Identity (1)
          Identity: host/iseuser1
    > AVP: t=Message-Authenticator(80) l=18 val=262b63190f7340d9b9db2f888ea1cb79
    > AVP: t=EAP-Key-Name(102) l=2 val=
```

액세스 챌린지

ISE가 액세스 요청을 수신하고 응답하기로 결정한다고 가정하면 이 응답은 ISE에서 액세스 챌린지로 올 수 있습니다. 액세스 요청을 다시 살펴보면 AVP가 시작되기 전에 RADIUS 패킷 ID가 36으로 표시됩니다.

WLC가 액세스 챌린지를 수신할 때 RADIUS ID는 액세스 요청의 해당 패킷 ID와 일치해야 합니다. RADIUS 패킷 ID는 ISE와 WLC 간의 RADIUS 통신을 위한 것입니다. 또한 이 패킷에서 ISE가 ISE와 클라이언트 간의 통신을 추적하는 데 사용되는 새로운 EAP ID인 201을 설정했음을 확인할 수 있습니다. 이 시점에서 WLC는 ISE와 클라이언트 간의 통신을 위한 패스스루일 뿐입니다.

여기서 이러한 모든 패킷 ID를 메모하여 통신 흐름 및 네트워크를 통해 이러한 패킷을 추적하는 방법을 이해하는 것이 중요합니다. 프로덕션 환경에서는 일반적으로 여러 인증이 동시에 발생합니다. 클라이언트의 MAC 주소에 패킷을 매칭하려면 calling-station-id 명령을 사용합니다. 그런 다음 RADIUS 패킷 ID 및 EAP 패킷 ID를 사용하여 이 특정 클라이언트에 대한 패킷 흐름을 추적할 수 있습니다. 지금까지 어느 쪽도 인증서를 보내지 않았으므로 MTU에 대해 걱정할 필요는 없습니다.

```
> Frame 285: 169 bytes on wire (1352 bits), 169 bytes captured (1352 bits)
> Ethernet II, Src: VMware_8c:8e:41 (00:0c:29:8c:8e:41), Dst: Cisco_56:49:8b (f4:bd:9e:56:49:8b)
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 260
> Internet Protocol Version 4, Src: 192.168.160.88, Dst: 192.168.160.20
> User Datagram Protocol, Src Port: 1812, Dst Port: 58038
v RADIUS Protocol
    Code: Access-Challenge (11)
    Packet identifier: 0x24 (36)
    Length: 123
    Authenticator: 9046d29958d0812d0a1cac17f20842a0
    [This is a response to a request in frame 281]
    [Time from request: 0.003997000 seconds]
  v Attribute Value Pairs
    > AVP: t=State(24) l=77 val=333743504d53657373696f6e49443d3134413041384330303030303030313041
    v AVP: t=EAP-Message(79) l=8 Last Segment[1]
        Type: 79
        Length: 8
        EAP fragment: 01c900060d20
      v Extensible Authentication Protocol
          Code: Request (1)
          Id: 201
          Length: 6
          Type: TLS EAP (EAP-TLS) (13)
        > EAP-TLS Flags: 0x20
    > AVP: t=Message-Authenticator(80) l=18 val=587539e3839e8a4eef6c6d5735443d3a
```

EAP 요청 및 EAP 응답

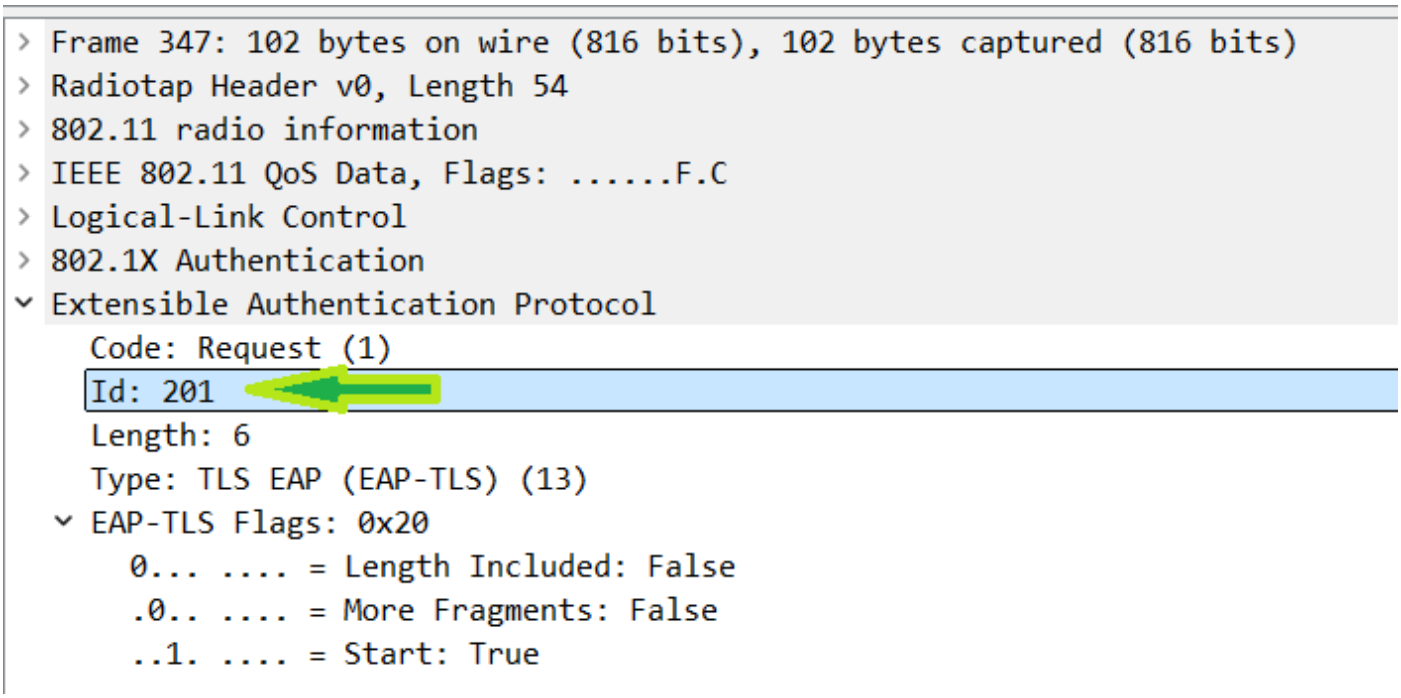
다시 한 번 말씀드리지만, 클라이언트는 RADIUS가 아닌 EAP를 사용합니다. 즉, WLC가 액세스 챌린지를 받을 때 RADIUS 데이터를 제거하고 EAP 요청을 당겨서 클라이언트에 보낼 수 있도록 해야 합니다.

EAP 요청

이는 WLC가 액세스 챌린지를 받았을 때의 내부 방법과 정확히 일치해야 합니다. 그러나 모든 RADIUS 항목이 제거되고 EAP 부분만 클라이언트로 전송됩니다.

WLC가 ISE에서 받은 것과 동일한 데이터이므로 액세스 챌린지에 있던 것처럼 여기에서 EAP 패킷 ID 201을 계속 볼 수 있습니다. 여기서의 흐름은 EAP-ID와 동일하며, 이제 WLC에서 나오지 않고

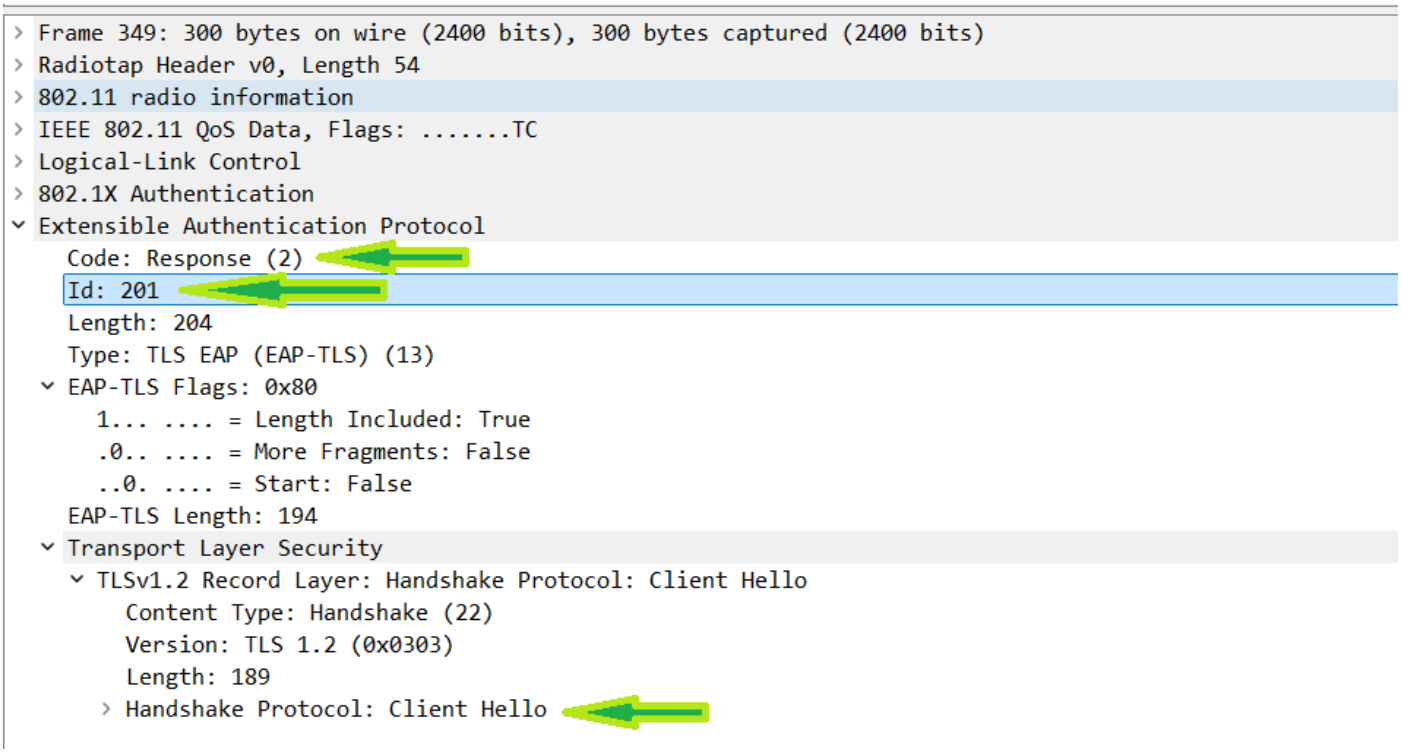
EAP 방법을 설정하는 데 사용됩니다. 이 패킷은 EAP-TLS 세션의 시작을 설정하기위한 것이므로 여전히 매우 작습니다.



EAP 응답

클라이언트가 EAP 요청을 받으면 EAP 응답으로 응답해야 합니다. EAP-Response에서 클라이언트는 TLS 세션을 설정하기 시작합니다. 이는 TLS가 사용되는 다른 상황과 동일하게 보입니다. "client hello" 메시지로 시작합니다. 이 문서는 이 주제와 무관하므로 클라이언트 hello에 어떤 내용이 포함되는지는 다루지 않습니다. 여기서 알아야 할 사항은 TLS 세션이 설정 중이라는 것입니다.

다른 TLS 설정과 마찬가지로 여기에서 패킷의 데이터를 볼 수 있습니다. EAP-ID 응답과 마찬가지로 이 패킷은 WLC에 도달하고 액세스 요청으로 변환됩니다. ISE는 액세스 챌린지에 패키징된 EAP 요청으로 응답합니다. 이것은 지금부터의 흐름으로 계속됩니다.



TLS 인증서

여기에서 패킷 크기가 증가하는 것을 확인할 수 있습니다. 하나 이상의 중간 CA(Certificate Authority)가 있는 경우 인증서가 상당히 클 수 있습니다. 자체 서명 인증서인 경우 2개의 중간 CA 및 루트 CA에 연결된 디바이스 인증서가 있는 인증서보다 작습니다. 어느 쪽이든 일반적으로 인증서 소유자가 여기에서 자체 패킷을 조각화하기 시작하는 것을 볼 수 있습니다.

ISE 인증서

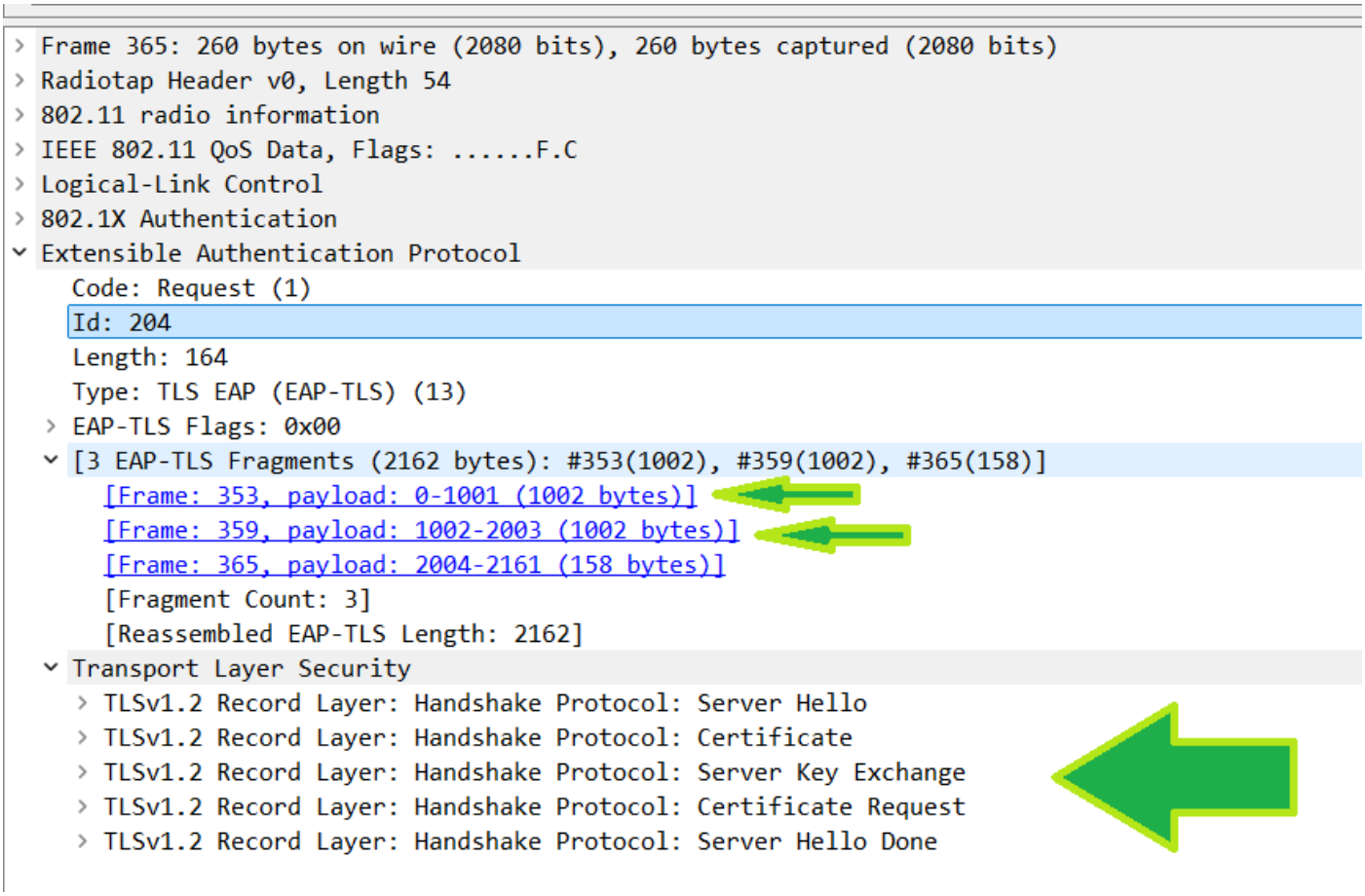
이제 ISE가 TLS 클라이언트 hello를 수신했으므로 다른 EAP 요청에 응답합니다. 이 새 EAP 요청 ISE는 한 번에 "서버 hello" 메시지, 인증서, "서버 키 교환", "인증서 요청" 및 "서버 hello done" 메시지를 보냅니다. 이 모든 것을 하나의 패킷으로 전송하면 패킷은 네트워크에 대한 MTU를 초과하게 됩니다. 따라서 ISE는 패킷 자체를 프래그먼트화하여 MTU에 따라 패킷을 가져옵니다. ISE에서는 패킷의 데이터 부분을 프래그먼트화하여, 이중 프래그먼트화를 방지하고자 1002바이트보다 크지 않도록 합니다.

이중 단편화는 무엇을 의미합니까? 전송하려는 데이터가 너무 커서 네트워크의 MTU에 맞지 않기 때문에 첫 번째 조각화가 ISE에서 발생하고 있습니다. 그러나 MTU가 동일하더라도 네트워크 설정 방식 때문에 디바이스가 헤더를 추가하고 MTU에 머물러 있으려면 패킷을 프래그먼트화해야 할 수 있는 다른 위치가 네트워크에 있을 수 있습니다. 이는 조각화 금지 비트가 선택된 경우에도 마찬가지일 수 있습니다.

VPN 터널 또는 이와 관련된 모든 터널을 사용하는 것이 좋은 예입니다. VPN 터널에 데이터를 넣으려면 VPN 라우터가 트래픽에 헤더를 추가해야 합니다. 이 RADIUS 트래픽이 MTU에서 또는 MTU에 가깝게 조각화된 경우, 이 VPN에서는 데이터를 MTU에 유지하고 추가 헤더를 추가할 방법이 없습니다. 이는 CAPWAP 터널에서도 마찬가지이며, 나중에 볼 수 있습니다.

따라서 이러한 패킷이 다른 디바이스에서 다시 프래그먼트화되는 상황을 피하려면 ISE는 대부분의

네트워크에서 이를 방지할 수 있는 위치에서 패킷을 프래그먼트화합니다. 즉, ISE는 매번 빈 EAP 응답을 대기 중인 여러 EAP 요청에서 이 데이터를 보냅니다. EAP ID는 전송되는 각 프래그먼트에 따라 증가합니다. WLC의 관점에서, 이것은 모든 프래그먼트에 대한 액세스 챌린지 및 액세스 요청 교환이 될 것이며, RADIUS 패킷 ID는 전송되는 각 프래그먼트와 함께 증가할 것이다.



클라이언트 인증서

ISE에서 모든 프래그먼트를 전송하고 클라이언트에서 프래그먼트를 리어셈블하면 패킷 흐름은 클라이언트로 이동하여 어떤 것을 전송합니다. TLS에서는 클라이언트에서 인증을 완료하기 위해 이 시점에 자체 인증서를 보내야 합니다. 여기서 일이 더 복잡해집니다. ISE와 마찬가지로 클라이언트는 여러 TLS 부품을 한꺼번에 전송하는데, 그중 하나가 인증서입니다.

ISE에서 볼 수 있었던 것과 달리, 대부분의 클라이언트는 MTU 바로 아래에 EAP 데이터를 보냅니다. 이 데모에서는 802.1x 데이터가 1492입니다. 문제는 AP가 WLC로 전송할 수 있도록 CAPWAP 헤더를 추가해야 한다는 것입니다.

어떻게 해야 할까요? AP는 헤더를 추가하고 WLC로 전송할 수 있도록 패킷을 프래그먼트화해야 합니다. AP가 패킷을 단편화하지 않고 WLC로 가져올 수 있는 방법은 없습니다. 즉, 여기서 패킷은 클라이언트로부터 먼저, 다시 AP로부터 이중 프래그먼트화됩니다. 그러나 CAPWAP에서 예상할 수 있듯이 이러한 단편화는 일반적으로 문제가 되지 않습니다.

무선 패킷:

```
> Frame 367: 1588 bytes (12704 bits), 1588 bytes captured (12704 bits)
> Radiotap Header v0, Length 54
> 802.11 radio information
> IEEE 802.11 QoS Data, Flags: .......TC
> Logical-Link Control
> 802.1X Authentication
v Extensible Authentication Protocol
    Code: Response (2)
    Id: 204
    Length: 1492
    Type: TLS EAP (EAP-TLS) (13)
  v EAP-TLS Flags: 0xc0
      1... .... = Length Included: True
      .1.. .... = More Fragments: True
      ..0. .... = Start: False
    EAP-TLS Length: 4692
```

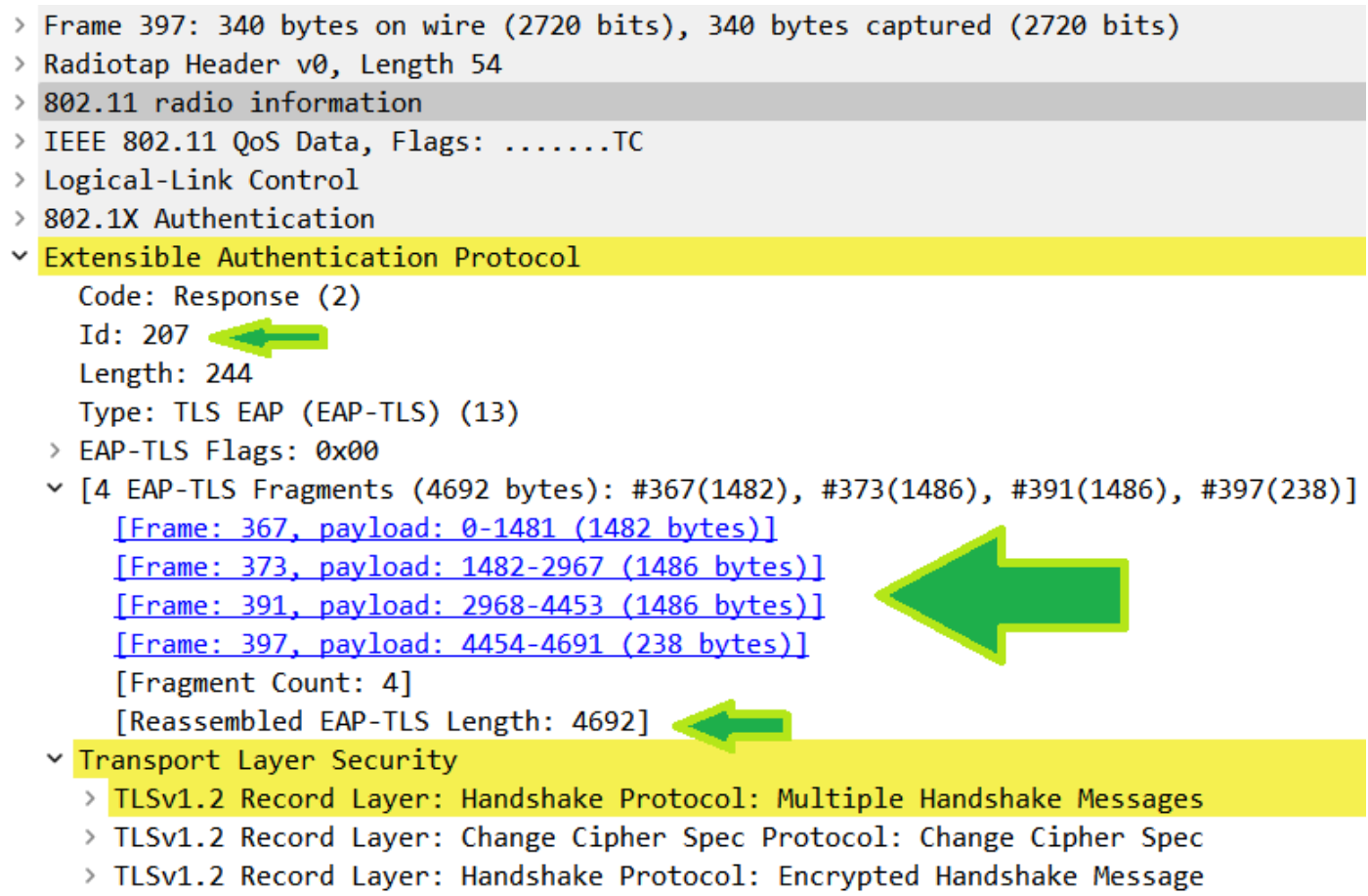
전선의 패킷 조각:

```
> Frame 56: 1482 bytes (11856 bits), 1482 bytes captured (11856 bits) on interface /tmp
> Ethernet II, Src: Cisco_b5:e6:00 (0c:75:bd:b5:e6:00), Dst: Cisco_56:49:8b (f4:bd:9e:56:49:8b)
> Internet Protocol Version 4, Src: 192.168.160.116, Dst: 192.168.160.20
> User Datagram Protocol, Src Port: 5248, Dst Port: 5247
> Control And Provisioning of Wireless Access Points - Data
  [Reassembled in: 57]
v Data (1424 bytes)
    Data: 018800000c75bdb3022038689362ec7e0c75bdb3022f00010000aaaa03000000888e0100…
    [Length: 1424]
```

와이어에서 패킷이 리어셈블되었습니다.

```
Wireshark · Packet 57 · FromTheWire2.pcap
> Frame 57: 156 bytes (1248 bits), 156 bytes captured (1248 bits) on interface /tmp/epc_ws/wif_to_ts_pipe, id 0
> Ethernet II, Src: Cisco_b5:e6:00 (0c:75:bd:b5:e6:00), Dst: Cisco_56:49:8b (f4:bd:9e:56:49:8b)
> Internet Protocol Version 4, Src: 192.168.160.116, Dst: 192.168.160.20
> User Datagram Protocol, Src Port: 5248, Dst Port: 5247
> Control And Provisioning of Wireless Access Points - Data
> [2 Message fragments (1530 bytes): #56(1424), #57(106)]
> IEEE 802.11 QoS Data, Flags: .......T
> Logical-Link Control
> 802.1X Authentication
v Extensible Authentication Protocol
    Code: Response (2)
    Id: 204
    Length: 1492
    Type: TLS EAP (EAP-TLS) (13)
  > EAP-TLS Flags: 0xc0
    EAP-TLS Length: 4692
```

모든 클라이언트 조각이 공중으로 리어셈블됨:



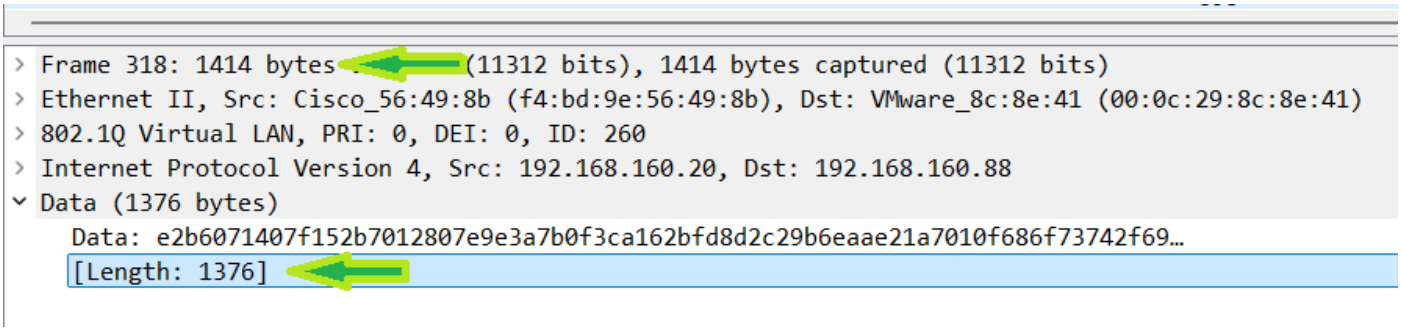
WLC의 클라이언트 인증서

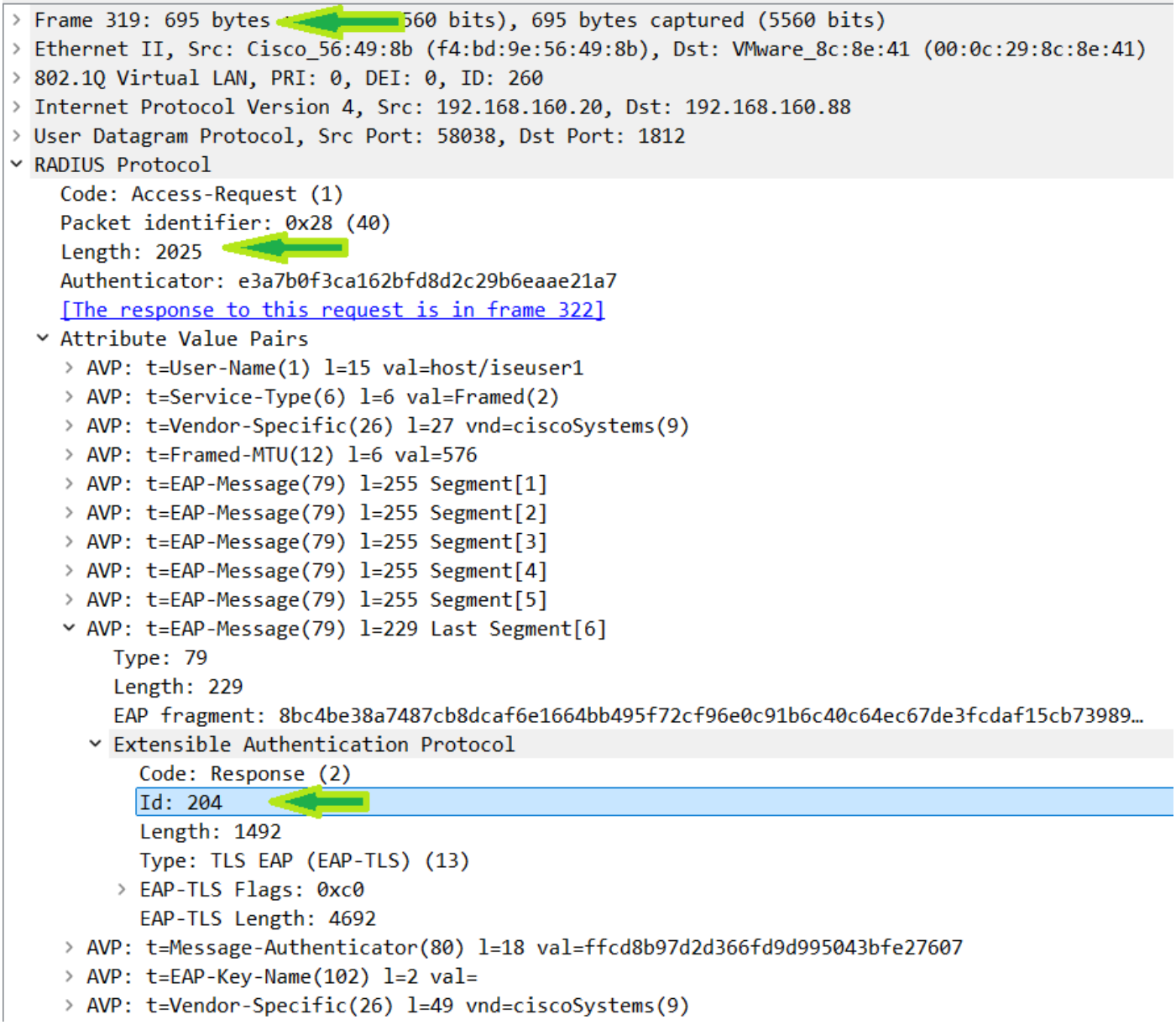
WLC는 2개의 CAPWAP 프래그먼트를 수신하고 클라이언트에서 전체 1492바이트 패킷을 포함하도록 리어셈블하며, 패킷을 복원하지만 오래 걸리지 않습니다. WLC가 액세스 요청을 전송하는 방법을 되돌아보면, ISE에 데이터를 보내기 전에 패킷에 약 400바이트의 RADIUS AVP를 추가해야 하므로 이 복원은 수명이 짧습니다.

간단한 계산을 위해 WLC에서 408바이트를 추가하여 총 패킷 크기를 1900으로 계산한다고 가정합니다. 이것은 1500 MTU를 훨씬 초과합니다. 그렇다면 WLC는 어떻게 해야 할까요? 패킷을 다시 프래그먼트화합니다.

이 시점에서 WLC는 기본적으로 1396에서 패킷을 프래그먼트화합니다. 여기의 생각은 ISE와 동일합니다. 패킷을 충분히 작게 만들어서 다른 터널을 통과해야 할 경우 헤더를 추가하기 위해 다시 프래그먼트화할 필요가 없도록 해야 합니다. 그러나 WLC는 ISE만큼 신중하지 않으므로 여기서는 1396으로 충분합니다.

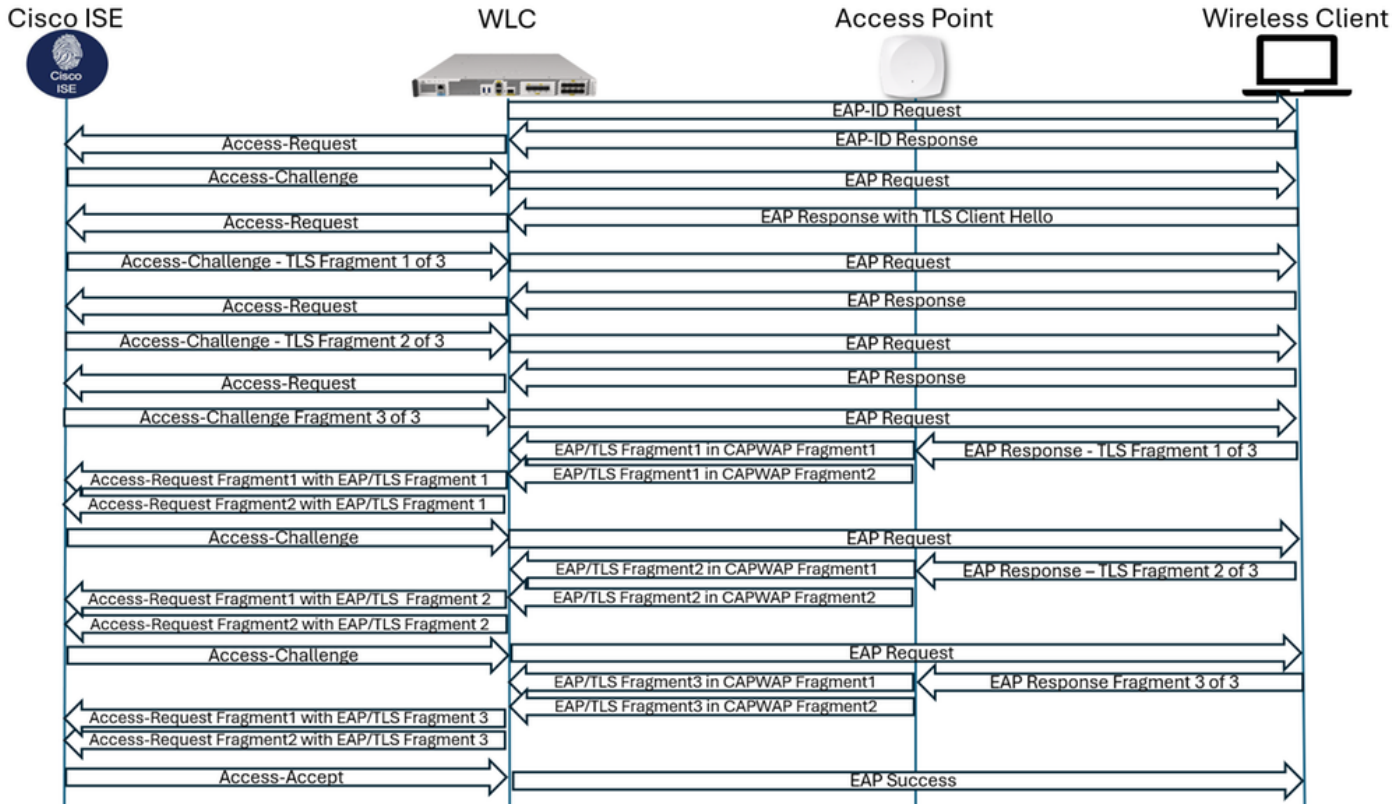
WLC를 떠나는 조각화된 패킷:





패킷 흐름 TL;DR

ISE가 인증서를 전송할 때 TLS 패킷을 1002바이트로 프래그먼트화합니다. 거기에는 아무 문제도 없습니다. 클라이언트가 인증서를 보낼 때 일반적으로 MTU에 가깝게 프래그먼트화됩니다. AP는 CAPWAP 헤더를 패킷에 추가해야 하므로 이 패킷도 프래그먼트화해야 합니다. WLc가 프래그먼트를 수신하면 패킷을 리어셈블해야 하지만, RADIUS AVP를 추가해야 패킷이 다시 프래그먼트화됩니다. 패킷 흐름은 다음과 같습니다.



RADIUS MTU 동작 변경

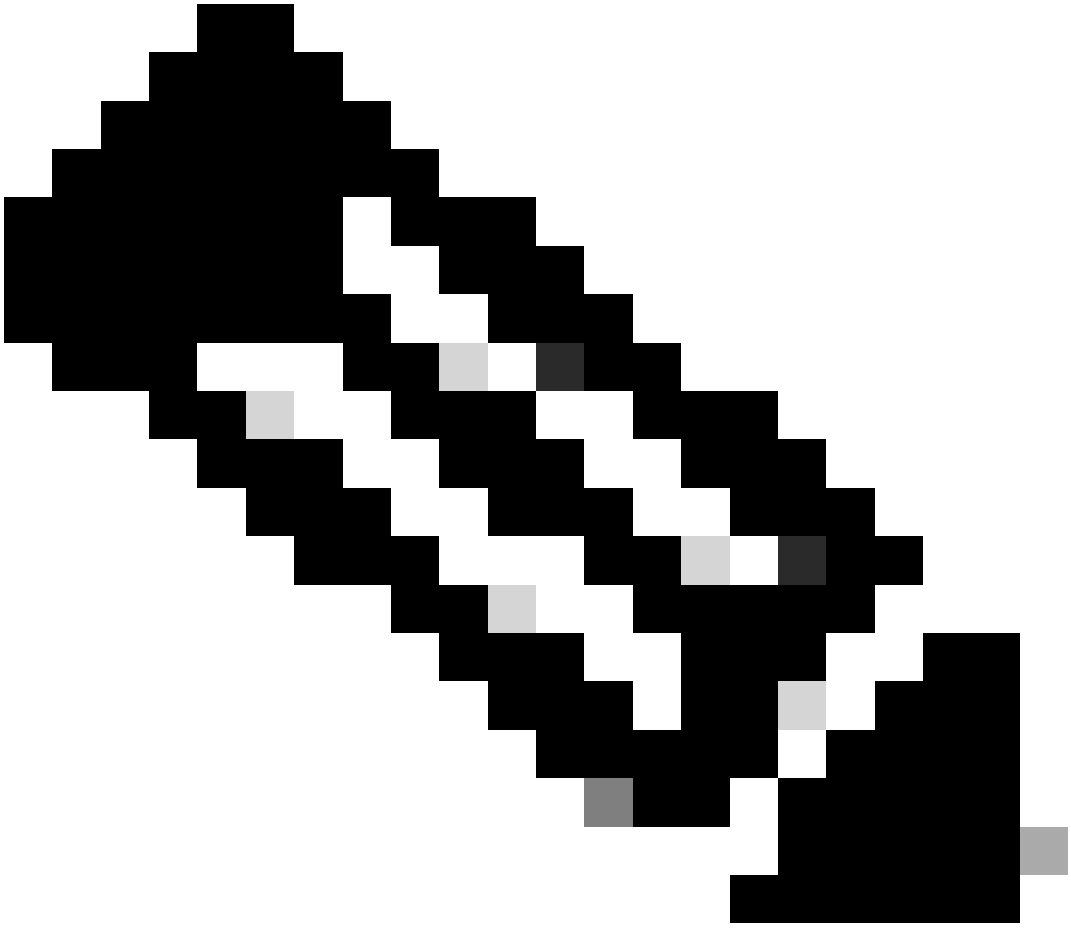
모든 무선 클라이언트 데이터 트래픽에 대한 패킷 흐름을 보면 무선 인프라가 몇 군데에서만 영향을 미친다는 것을 알 수 있습니다. 첫 번째는 트래픽이 AP를 떠날 때이고 두 번째는 트래픽이 WLC를 떠날 때입니다. 무선 인프라에서 클라이언트 MSS를 조정할 수 있는 TCP 트래픽은 예외입니다. 그러나 EAP는 TCP에 해당하지 않으며, 실제로 자체 프로토콜입니다.

EAP 및 RADIUS 트래픽 흐름을 보면 원래 패킷 크기가 MTU에 너무 가까워지면 네트워크가 AP와 WLC의 트래픽 크기에 실제로 영향을 미친다는 것을 알 수 있습니다. 이 교환에서 WLC의 역할을 제대로 이해하면 RADIUS 패킷의 크기에 영향을 주는 곳이 실제로 한 곳뿐임을 알 수 있습니다. 이는 EAP 응답이 오고 RADIUS 액세스 요청으로 변경 할 때 입니다.

변경 사항

EAP 응답이 MTU를 초과할 경우, RADIUS AVP를 추가하면 프래그먼트화해야 합니다. 이 패킷은 반드시 프래그먼트화해야 하므로 프래그먼트화할 크기를 적어도 결정할 수 있습니다. 17.11에 도입된 동작 변경이 적용되는 부분입니다.

Cisco 버그 ID CSCwc81849에서 추적한 기능 요청에서 RADIUS 점보 패킷에 대한 지원을 추가하려고 합니다. 이렇게 하면 RADIUS 패킷이 1396에서 더 이상 자동으로 프래그먼트화되지 않습니다. 이제 ip radius source-interface <interface X> 명령을 추가하면 RADIUS access-request가 해당 인터페이스의 MTU에서 전송됩니다.

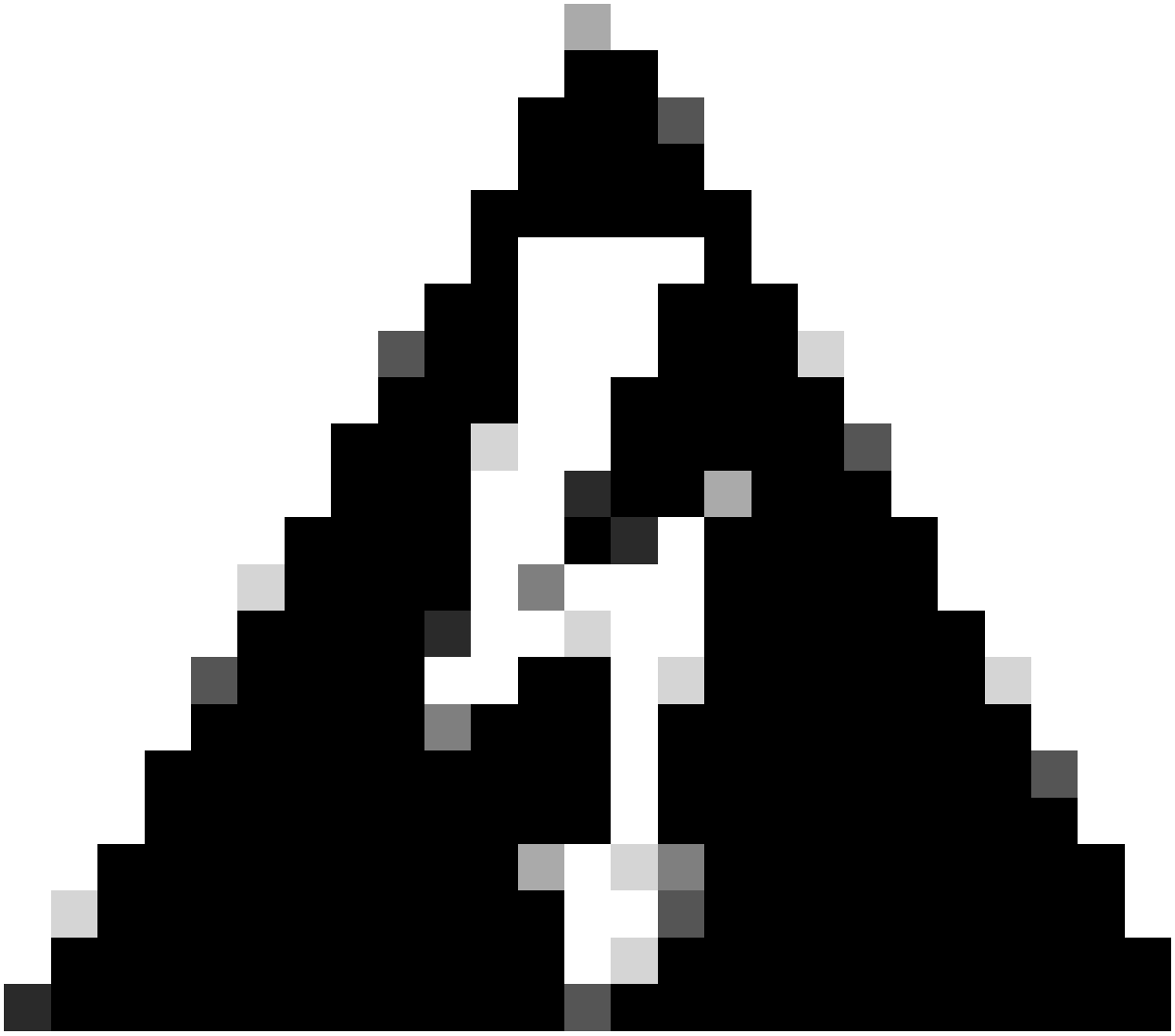


참고: Cisco Catalyst Center를 사용하는 경우 AAA 구성을 프로비저닝하면 소스 인터페이스가 서버 그룹에 자동으로 추가됩니다. 그러면 기본 동작이 해당 명령에 사용된 인터페이스의 MTU 크기로 프래그먼트화됩니다.

이 변경 사항을 어떻게 사용할 수 있습니까?

모든 인터페이스의 기본 MTU는 1500이므로, 프래그먼트할 새 MTU가 됩니다. 모든 RADIUS 트래픽에 사용되는 기본 인터페이스는 WMI(Wireless Management Interface)입니다. 서버 그룹의 컨피그레이션을 볼 때 소스 인터페이스가 지정되지 않은 경우 WLC는 WMI를 사용하여 1396에서 RADIUS 트래픽을 전송합니다. 그러나 서버 그룹 컨피그레이션으로 이동하여 소스 인터페이스가 WMI라고 말하면 이제 WLC에서 WMI를 사용하여 1500에서 RADIUS 트래픽을 전송합니다.

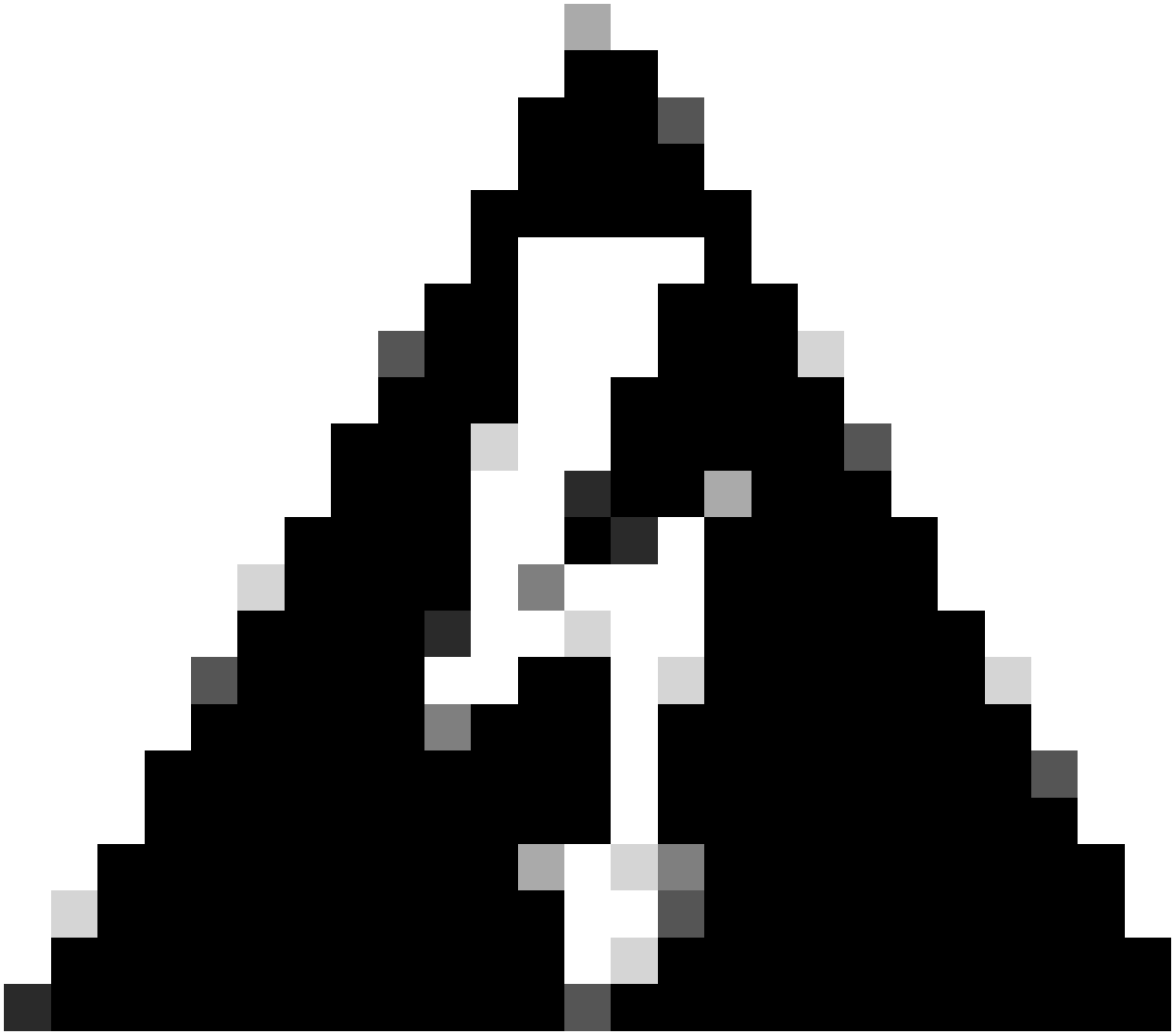
이제 앞서 설명한 VPN과 같은 디바이스가 네트워크에 있다고 가정합니다. 트래픽을 이중 프래그먼트화하지 않도록 하여 인터페이스의 MTU를 더 작은 값으로 변경하여 다른 위치에서 패킷을 프래그먼트화할 수 있습니다. 패킷이 1500이 아닌 1200바이트 표시에서 프래그먼트화되도록 MTU를 1200과 같은 형식으로 변경할 수 있습니다.



경고: WMI의 MTU를 변경하면 WLC 관리 IP 주소를 오가는 모든 트래픽에 영향을 줍니다.

WMI의 MTU를 변경하지 않으려는 경우에도 소스 인터페이스를 지정하는 핵심은 WMI에서 다른 인터페이스로 변경하고 RADIUS 트래픽에 해당 인터페이스를 사용하고 해당 인터페이스의 MTU를 변경하는 것입니다. 이 컨피그레이션은 서버 그룹 레벨에서 수행되므로, 어떤 RADIUS 트래픽의 영향을 받을지 매우 구체적으로 지정할 수 있습니다.

이 컨피그레이션은 AAA 서버 또는 WLAN에 연결되어 있지 않습니다. 동일한 서버가 포함된 여러 서버 그룹을 가질 수 있으며, 이 중 하나를 선택하는 경우에만 소스 인터페이스를 지정할 수 있습니다. 이 서버 그룹은 방법 목록에 추가된 다음 WLAN에 추가됩니다. 따라서 예를 들어, WLAN을 하나만 변경할 경우, AAA 서버가 하나뿐이더라도 새 서버 그룹을 만들고, 사용할 MTU가 있는 인터페이스를 가리키는 ip radius source-interface 명령을 사용하고, AAA 서버를 이 새 그룹에 추가하고, 이 새 그룹을 사용하여 새 방법 목록을 만든 다음, 이 방법 목록을 변경할 특정 WLAN에 추가할 수 있습니다.



경고: 라이브 네트워크를 변경할 때는 유지 보수 기간 중에 수행하는 것이 좋습니다.

패킷 캡처에 증거가 있습니다.

흔히 알고 있는 사실이다네트워킹 분야에서 이를 캡처하지 않았다면 증명할 수 없습니다. 다음은 이러한 변경 사항을 적용한 몇 가지 컨피그레이션 예입니다.

WLAN 컨피그레이션입니다. 테스트 중에 메서드 목록에 사용되는 서버 그룹만 변경됩니다.

```
9800#show run wlan
wlan TLS-Test 2 TLS-Test
 radio policy dot11 24ghz
 radio policy dot11 5ghz
 no security ft adaptive
 security dot1x authentication-list TLS-AuthC
 no shutdown
!
```

기본 MTU를 사용하여 Source-Interface 명령 추가

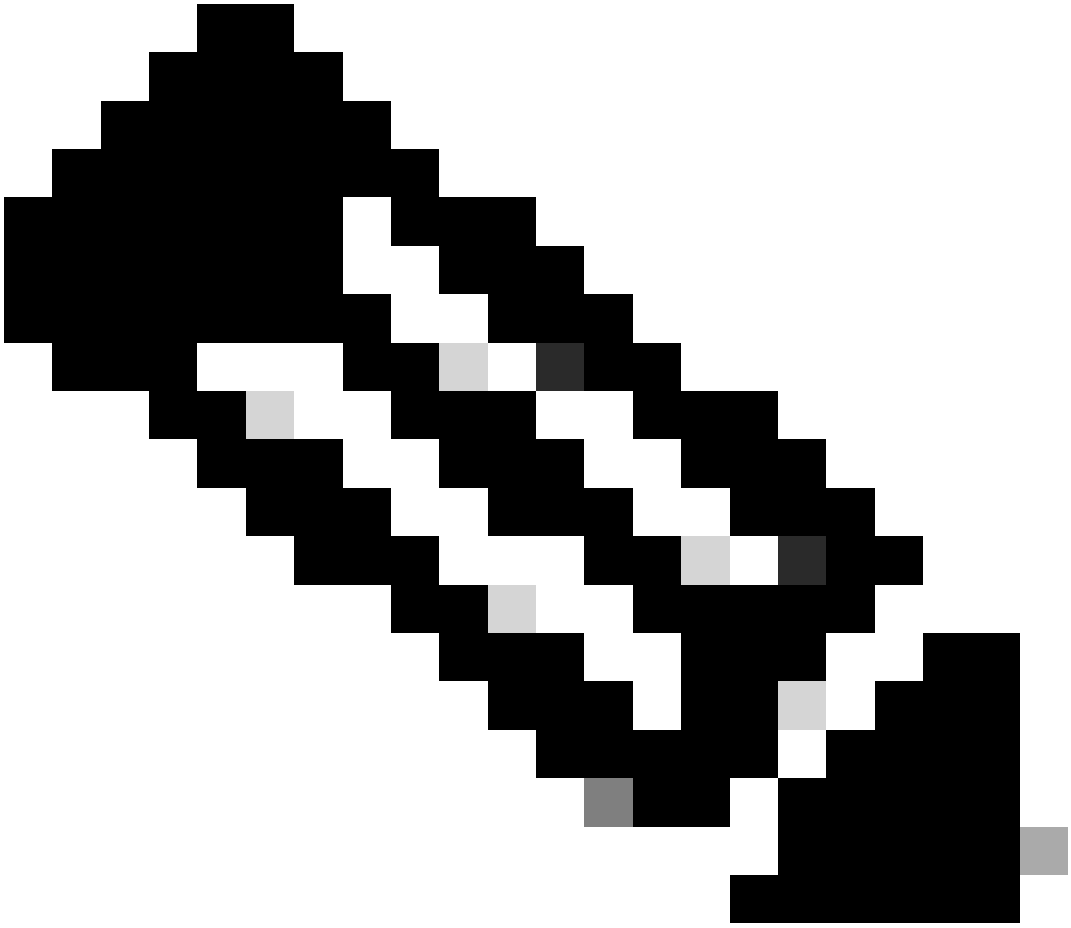
여기서는 ISE 서버를 가리키는 일반적인 서버 그룹일 뿐입니다. MTU가 설정되지 않은 WMI를 가리키는 source interface 명령이 추가되었습니다. 컨피그레이션은 다음과 같습니다.

```
9800#show run aaa
!
aaa authentication dot1x TLS-AuthC group NoMTU
!
!
radius server ISE
 address ipv4 192.168.160.10 auth-port 1812 acct-port 1813
 key 6 _`gINMNXObF[^bPBVNaYibbBMhNMFAbKUAAB
!
aaa group server radius NoMTU
 server name ISE
 ip radius source-interface Vlan260
 deadtime 5
!
9800#show run inter vlan 260
!
interface Vlan260
 ip address 192.168.160.20 255.255.255.0
 no ip proxy-arp
end
```

보시다시피 NoMTU 서버 그룹이 WLAN에 연결된 인증 방법 목록에 추가되었습니다. ip radius source-interface VLAN260 명령은 이 서버 그룹에 사용되며 VLAN 260은 MTU를 지정하지 않습니다. 이는 MTU가 1500을 사용함을 의미합니다. MTU가 1500인지 확인하기 위해 show run all 명령을 사용하여 출력에서 인터페이스를 찾을 수 있습니다.

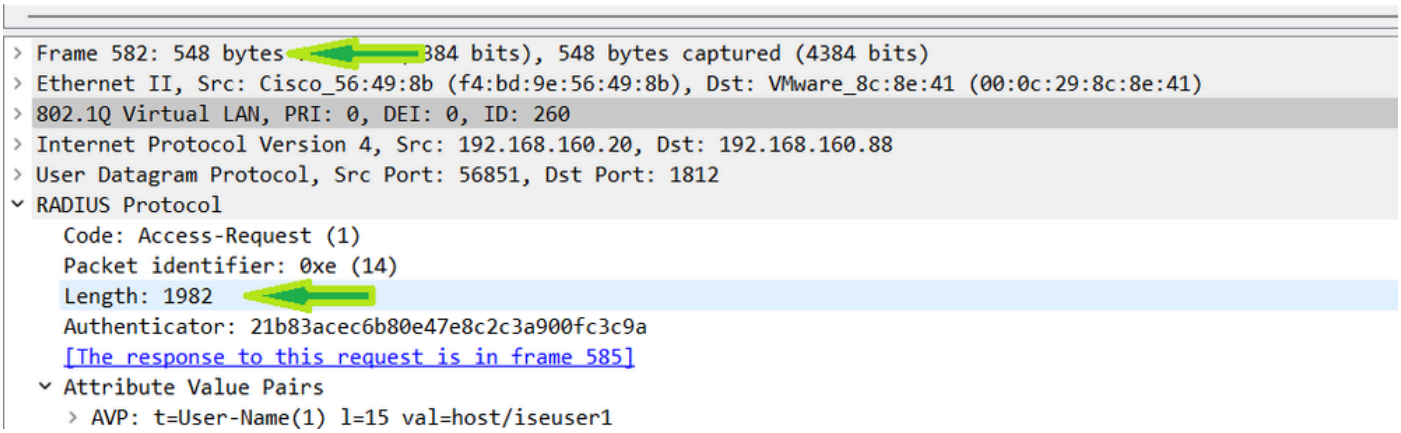
```
interface Vlan260
 ip address 192.168.160.20 255.255.255.0
 no ip clear-dont-fragment
 ip redirects
 ip unreachables
 no ip proxy-arp
 ip mtu 1500
```

이제 WLC에서 RADIUS 데이터를 추가하면 클라이언트 인증서가 ISE로 전송되어야 하는 패킷을 확인합니다.



참고: 여기서 행의 바이트는 1518입니다. 이것은 VLAN 헤더와 레이어 2 헤더처럼 이더넷 페이로드 외부에 있는 헤더들을 포함하고 있습니다. 이는 MTU에 포함되지 않습니다.

```
> Frame 581: 1518 bytes (12144 bits), 1518 bytes captured (12144 bits)
> Ethernet II, Src: Cisco_56:49:8b (f4:bd:9e:56:49:8b), Dst: VMware_8c:8e:41 (00:0c:29:8c:8e:41)
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 260
> Internet Protocol Version 4, Src: 192.168.160.20, Dst: 192.168.160.88
v Data (1480 bytes)
    Data: de13071407c63226010e07be21b83acec6b80e47e8c2c3a900fc3c9a010f686f73742f69…
    [Length: 1480]
```



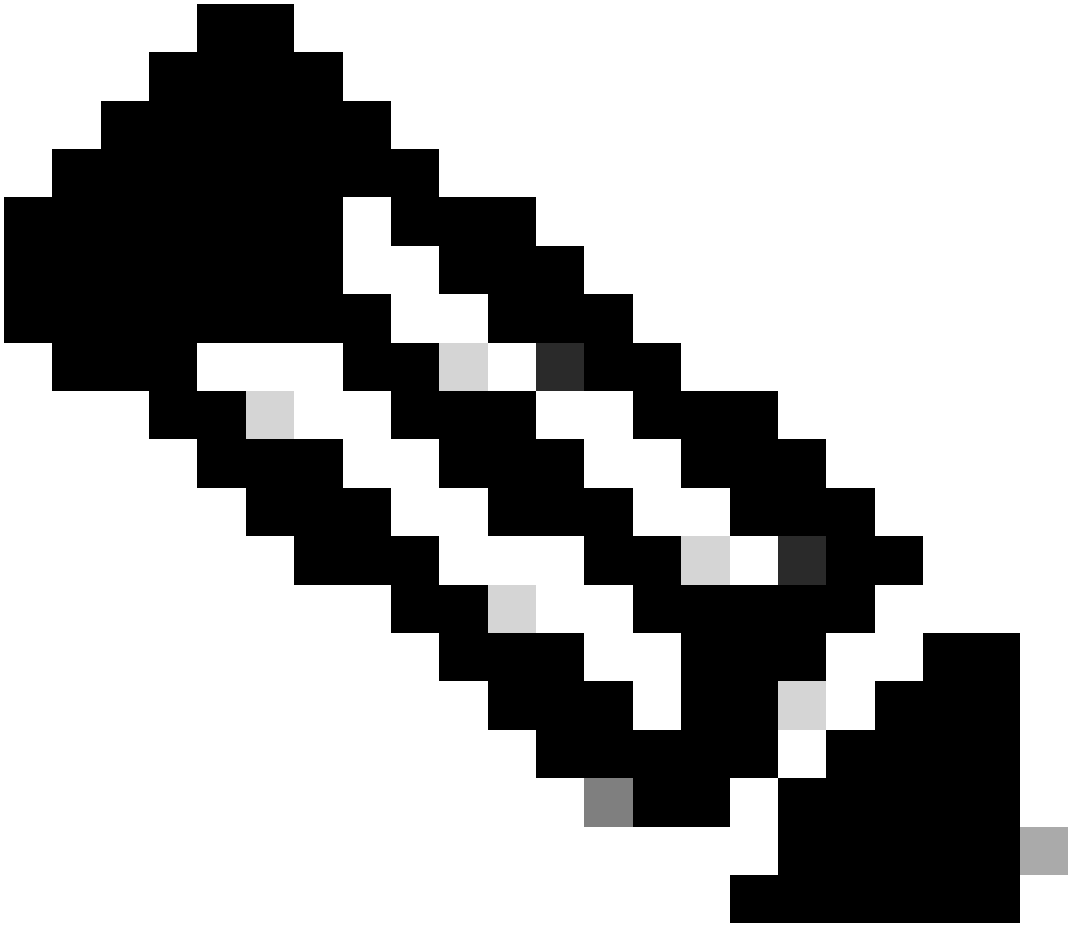
여기서, 당신은 1480에서 데이터 부분이 조각화된 것을 볼 수 있습니다. WMI의 1500 MTU에서 해당 프래그먼트를 가져올 수 있습니다. 다음 패킷은 550 바이트 이하이지만 RADIUS 데이터의 총 크기는 1982 입니다. 새로운 MTU로 프래그먼트화하면 이제 작동합니다.

MTU가 1200인 비 WMI 인터페이스 사용

이제 더 작은 MTU에서 프래그먼트화하되, 이 변경이 다른 트래픽에 영향을 미치지 않도록 한다고 가정합니다. 여기서는 문제가 없습니다. 컨피그레이션은 소스 인터페이스 컨피그레이션이 이 용도로 생성된 SVI를 가리키는 것과 동일하게 유지됩니다. 이 새 서버 그룹을 가리키도록 메서드 목록을 변경합니다. 이 서버 그룹은 내 WMI가 아닌 소스 인터페이스를 사용하며 MTU가 1200으로 설정되어 있습니다. 구성은 다음과 같습니다.

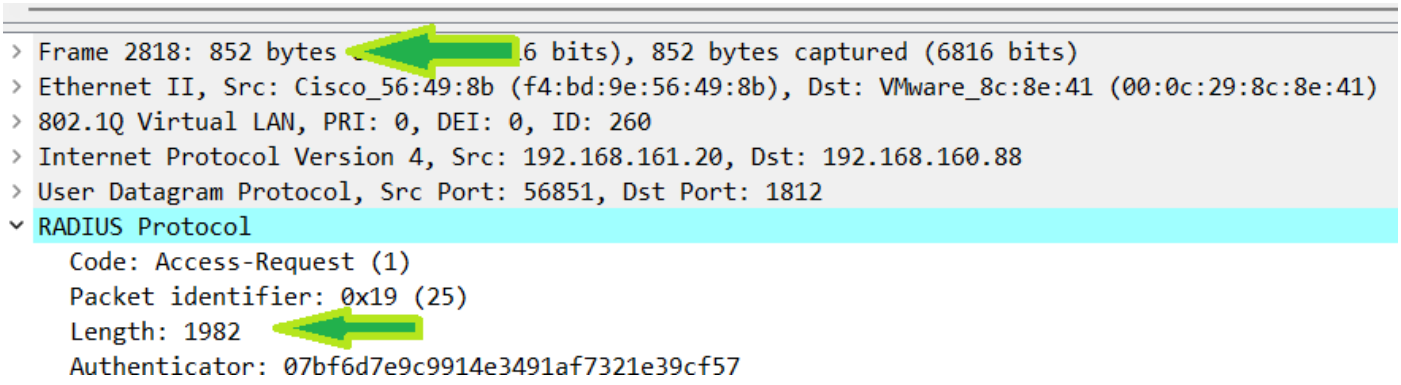
```
9800#show run aaa
!
aaa authentication dot1x TLS-AuthC group MTU1200
!
!
radius server ISE
 address ipv4 192.168.160.10 auth-port 1812 acct-port 1813
 key 6 _`gINMNXObFibbBMhNMFAbKUAAB
!
aaa group server radius MTU1200
 server name ISE
 ip radius source-interface Vlan261
 deadtime 5
!
9800#show run inter vlan 261
!
interface Vlan261
 ip address 192.168.161.20 255.255.255.0
 no ip proxy-arp
 ip mtu 1200
end
```

다음으로, 이 낮은 MTU로 패킷이 어떻게 표시되는지 알아보십시오.



참고: MTU를 낮추고 프래그먼트화 포인트를 변경하는 것은 새로운 동작의 일부가 아닙니다. 이것은 항상 사실이었다. 1396에서 프래그먼트화하는 기본 동작이 MTU에 맞지 않으면 항상 다른 지점에서 프래그먼트화됩니다. 이 섹션에서는 제공되는 옵션에 대해 설명합니다.

```
> Frame 2817: 1214 bytes [illegible]2 bits), 1214 bytes captured (9712 bits)
> Ethernet II, Src: Cisco_56:49:8b (f4:bd:9e:56:49:8b), Dst: VMware_8c:8e:41 (00:0c:29:8c:8e:41)
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 260
> Internet Protocol Version 4, Src: 192.168.161.20, Dst: 192.168.160.88
v Data (1176 bytes)
    Data: de13071407c6b995011907be07bf6d7e9c9914e3491af7321e39cf57010f686f73742f69…
    [Length: 1176]
```



여기서 RADIUS 데이터는 여전히 1982바이트이지만 이번에는 소스 인터페이스가 사용되지 않을 경우 데이터가 프래그먼트화되었을 기본 1376이 아니라 1176에서 프래그먼트화되었습니다. MTU를 1500으로 설정하고 source-interface 명령을 사용하면 1480에서 프래그먼트화됩니다. 여기서 컨피그레이션을 사용하면 WLC의 다른 트래픽을 방해하지 않고 더 낮은 MTU로 트래픽을 조작할 수 있습니다.

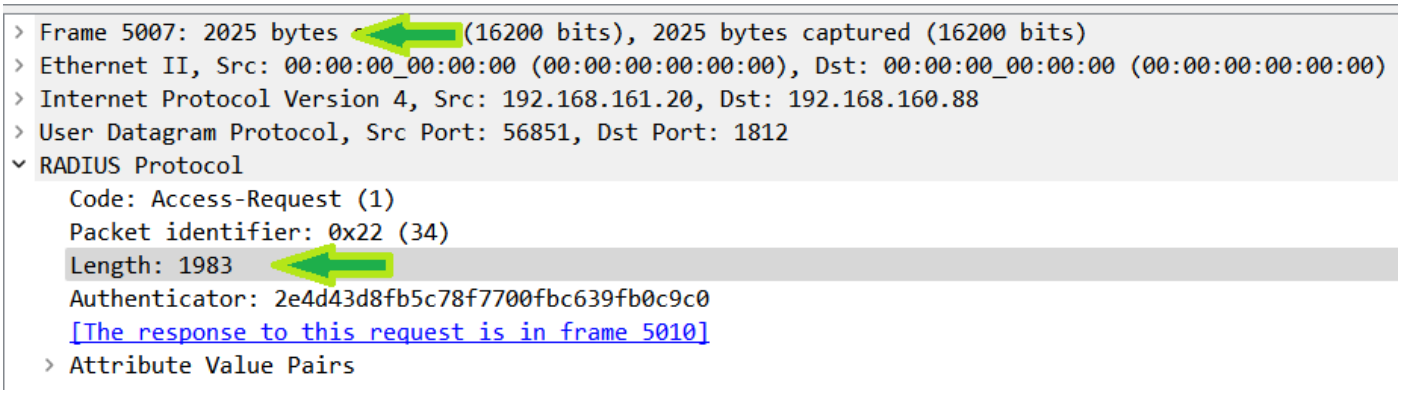
점보 프레임에 MTU 9000 사용

이 기능은 점보 프레임을 전송하는 옵션을 위해 만들어졌으므로 VLAN 261의 비 WMI 인터페이스를 계속 사용하지만 이를 테스트하지 않는 것이 아쉽습니다. 그러나 이제 IP MTU는 9000으로 설정됩니다. SVI에서 IP MTU를 설정하려면 MTU를 IP MTU보다 높게 설정해야 합니다. 이 컨피그레이션에서 이를 확인할 수 있습니다.

```
9800(config-if)#do sho run inter vl 261
!
interface Vlan261
 mtu 9100
 ip address 192.168.161.20 255.255.255.0
 no ip proxy-arp
 ip mtu 9000
end
```

여기에서 캡처를 보면 패킷이 절대 프래그먼트화되지 않았음을 알 수 있다. 이는 1983년에 RADIUS 데이터 크기의 단일 전체 패킷으로 전송되었습니다. 이 기능이 작동하려면 이 크기의 패킷이 통과할 수 있도록 네트워크의 나머지를 구성해야 합니다.

여기서 주의할 또 다른 점은 클라이언트 MTU가 변경되지 않았으므로 클라이언트가 여전히 1492에서 EAP 패킷을 프래그먼트화하고 있다는 것입니다. 차이점은 WLC가 클라이언트 데이터를 프래그먼트화하지 않고도 ISE에 패킷을 전송하는 데 필요한 모든 RADIUS 데이터를 추가할 수 있다는 것입니다.



결론

EAP-TLS를 사용하는 경우 클라이언트는 해당 인증서를 AAA 서버에 보내야 합니다. 이러한 인증서는 일반적으로 MTU보다 크므로 클라이언트가 프래그먼트화해야 합니다. 클라이언트가 데이터를 조각하는 지점은 MTU에 매우 가깝습니다. AP가 CAPWAP 헤더를 추가해야 하므로 클라이언트가 전송하는 것은 조각화되어야 합니다. WLC는 이 두 패킷을 받아 다시 결합하지만 RADIUS 데이터를 추가하기 위해 다시 프래그먼트화해야 합니다. 이 시점에서 네트워크 관리자는 클라이언트가 전송한 EAP 패킷을 WLC가 프래그먼트화하는 방법을 제어할 수 있습니다.

ip radius source-interface <interface you want to use> 명령을 AAA 서버 그룹에 추가하면 WLC는 WMI가 아닌(또는 WMI를 포함하여) 여기에 설치한 모든 인터페이스를 사용합니다. 또한 이 명령을 사용하면 WLC에 해당 인터페이스의 MTU가 기본값 1396 대신 어떤 것이든 프래그먼트화하도록 지시합니다. 이렇게 하면 네트워크를 통해 패킷이 이동하는 방식을 보다 효과적으로 제어할 수 있습니다.

Cisco Catalyst Center를 사용할 경우 source interface 명령이 서버 그룹에 추가되어 기본 동작이 변경됩니다.

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.