9800 WLCでのRADIUS MTUおよびフラグメンテーションについて

内容

はじめに

前提条件

- 要件
- 使用するコンポーネント

背景

9800 RADIUS MTU

EAP-TLSパケットフロー

- EAP-ID（オプション）
  - EAP-ID要求
  - EAP-ID応答
- Access-RequestおよびAccess-Challenge
  - アクセス要求
  - アクセスチャレンジ
- EAP要求とEAP応答
  - EAP要求
  - EAP応答
- TLS証明書
  - ISE証明書
  - クライアント証明書
  - WLCでのクライアント証明書
- パケットフローTL;DR

RADIUS MTU動作の変更

- 変更内容
- この変更の使用方法
- 証明はパケットキャプチャにあります
  - デフォルトのMTUを使用したSource-Interfaceコマンドの追加
  - Mtuが1200の非WMIインターフェイスを使用する
  - ジャンボフレームに9000のMTUを使用

結論

はじめに

このドキュメントでは、WLCがRADIUSサーバに送信するRADIUSパケットのMTUを設定する方法について説明します。

前提条件

要件

次の項目に関する基本的な知識があることが推奨されます。

- 9800ワイヤレスLANコントローラ(WLC)のAAA設定
- 認証、許可、アカウンティング(AAA)RADIUSの概念
- 拡張認証プロトコルEAP
- 最大伝送ユニット（MTU）

使用するコンポーネント

- Cisco Identity Service Engineer(ISE)3.2
- Catalyst 9800 Wireless Controllerシリーズ(Catalyst 9800-L)
- Cisco IOS® XE 17.15.2
- Windows 11ワイヤレスクライアント

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景

このドキュメントの目的に従い、使用するRemote Authentication Dial-In User Service(RADIUS)サーバはCisco ISEです。最初に、拡張認証プロトコル(EAP)プロセス中に外部の介入なしでパケットがどのように流れるかを示します。次に、WLCが任意のRADIUSサーバに送信するアクセス要求のサイズを変更する設定オプションを示します。このオプションは、IOS XEバージョン17.11で追加されました。

9800 RADIUS MTU

通常、RADIUSパケットのMTUは通常は小さく、いずれにしてもMTUに到達しないため、問題にはなりません。ただし、一方の側が証明書を送信する必要がある場合（通常は2 ~ 5 KB）、デバイスは証明書をフラグメント化してMTUの範囲内に収める必要があります。

EAP Transport Layer Security(EAP-TLS)の場合のように、クライアントが証明書をRADIUSサーバに送信する必要がある場合、WLCには、送信する必要があるRADIUSデータの量に応じてパケットのフラグメント化が再度必要になる状況が発生します。17.11までは、ネットワーク管理者はこのプロセスをほとんど制御できませんでしたが、現在では、エンジニアはWLCによって送信されるアクセス要求のサイズを操作するオプションを使用できます。

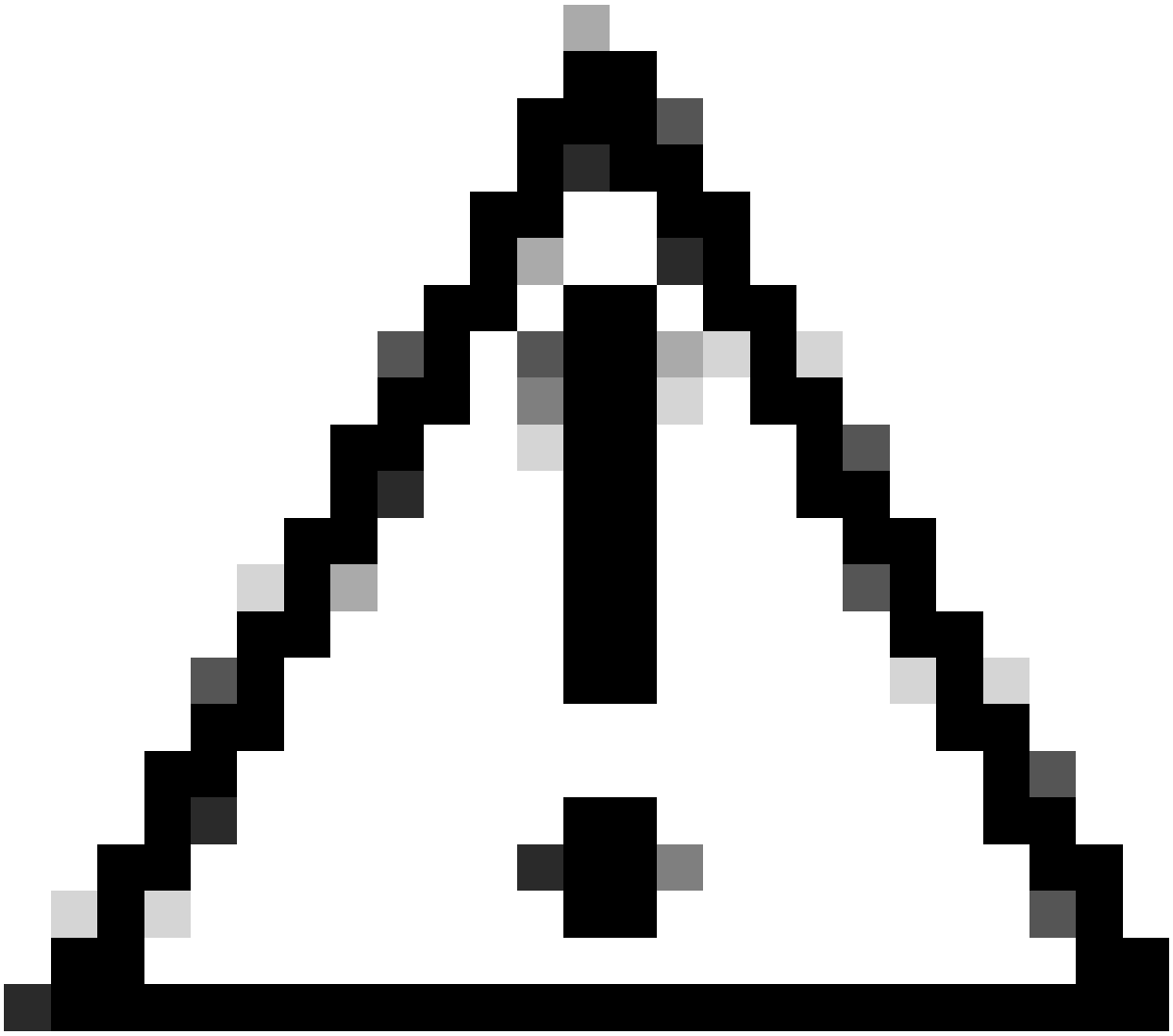
EAP-TLSパケットフロー

これは、パケットがどのように見え、ワイヤレスインフラストラクチャでどのように扱われるかについての詳細な説明です。 このドキュメントで紹介されている変更を完全に理解するには、dot1x、より具体的にはEAP-TLSを使用する場合のワイヤレス認証プロセス中のパケットフローを知ることが重要です。

EAPおよびRADIUSパケットフローがCiscoワイヤレスインフラストラクチャでどのように機能するかについて、すでに十分に理解している場合は、「動作の変更」の項に進んでください。この項では、17.11で追加された機能について説明し、ネットワーク管理者がRADIUS MTUをより詳細に制御できるようにします。最初に、EAP ID(EAP-ID)を確認します。

EAP-ID（オプション）

EAP-IDはオーセンティケータ（この場合はWLC）によって開始されます。これは、EAPプロセスの最初の部分である必要があります。ワイヤレスクライアントがEAPOL-Startを送信する場合があります。これは通常、クライアントがEAP-ID要求を受信しなかったか、最初からやり直すことを意味します。



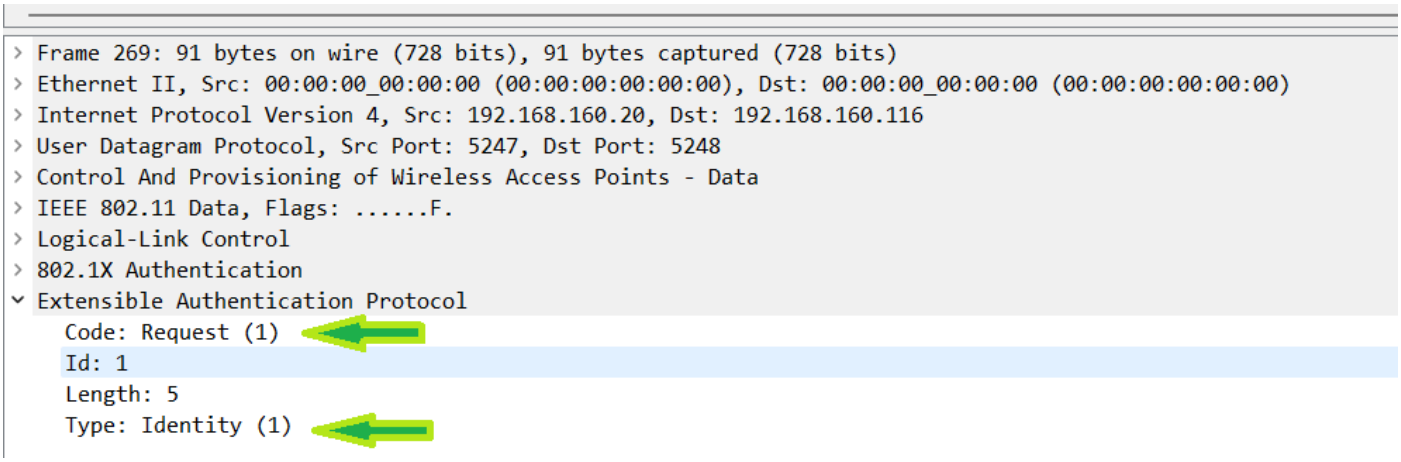
注意：EAP-IDパケットとEAPパケットIDには違いがあります。EAP-IDパケットは、サプリカントの識別に使用されます。ここで、EAPパケットIDは、ネットワークを移動する特定のパケットを追跡するために使用される番号です。

EAP-ID要求

まず、無線クライアントデバイスは、通常のアソシエーションプロセスを使用してネットワークに接続します。無線ローカルエリアネットワーク(WLAN)がdot1x用に設定されている場合、WLCはRADIUSサーバからのアクセスを要求する前に、クライアントが誰であるかを知る必要があります。この情報を見つけるために、WLCはクライアントとEAP-ID要求を送信します。

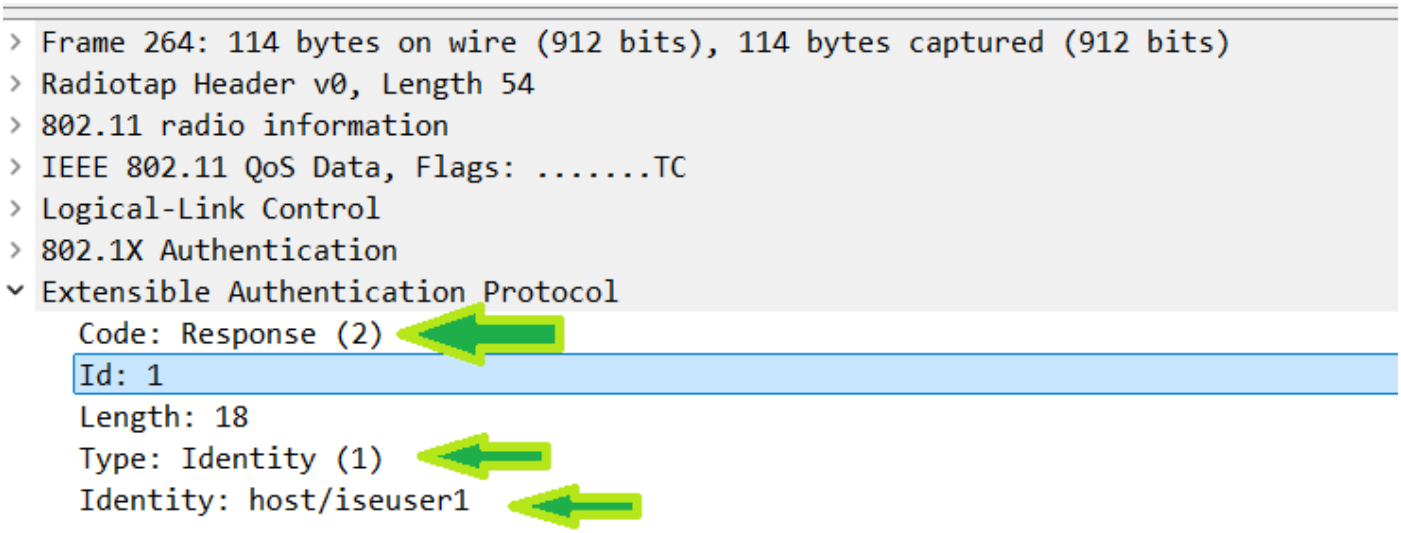
クライアントはEAP-ID応答で応答することが想定されます。これにより、アクセス要求を作成してISEに送信するために必要な情報がWLCに提供されます。EAP-ID要求は、通常のPEAP認証でクライアントにユーザ名とパスワードの入力を求める場合に発行されます。

ただし、この説明はEAP-TLSに関するものなので、ここでのEAP-ID応答にはユーザIDのみが含まれます。デモでは、ユーザIDはiseuser1です。このパケットでは、WLCがワイヤレスクライアントにEAP-ID要求を送信し、クライアントの身元を尋ねていることがわかります。これはワイヤレスクライアントであるため、WLCは要求をCAPWAPにカプセル化し、アクセスポイント(AP)に送信して無線で送信します。EAPデータでは、コード1は要求であることを示し、タイプ1はアイデンティティ用であることを示します。



EAP-ID応答

次に、ワイヤレスクライアントがEAP-ID応答で応答することが想定されます。EAPデータでは、コードは応答であることを示す2に変更されていますが、タイプは1のままであり、ID用であることを示しています。ここでは、クライアントが使用しているユーザ名も確認できます。これらのパケットで確認すべきもう1つの事項は、EAPパケットのID番号です。EAP-ID交換の場合は常に1ですが、ISEが関与するとこの番号は後で他の番号に変更されます。



両方のパケットはかなり小さいことがわかります。したがって、MTUはネットワークで使用される1500バイトより十分小さいため、ここでは関係ありません。

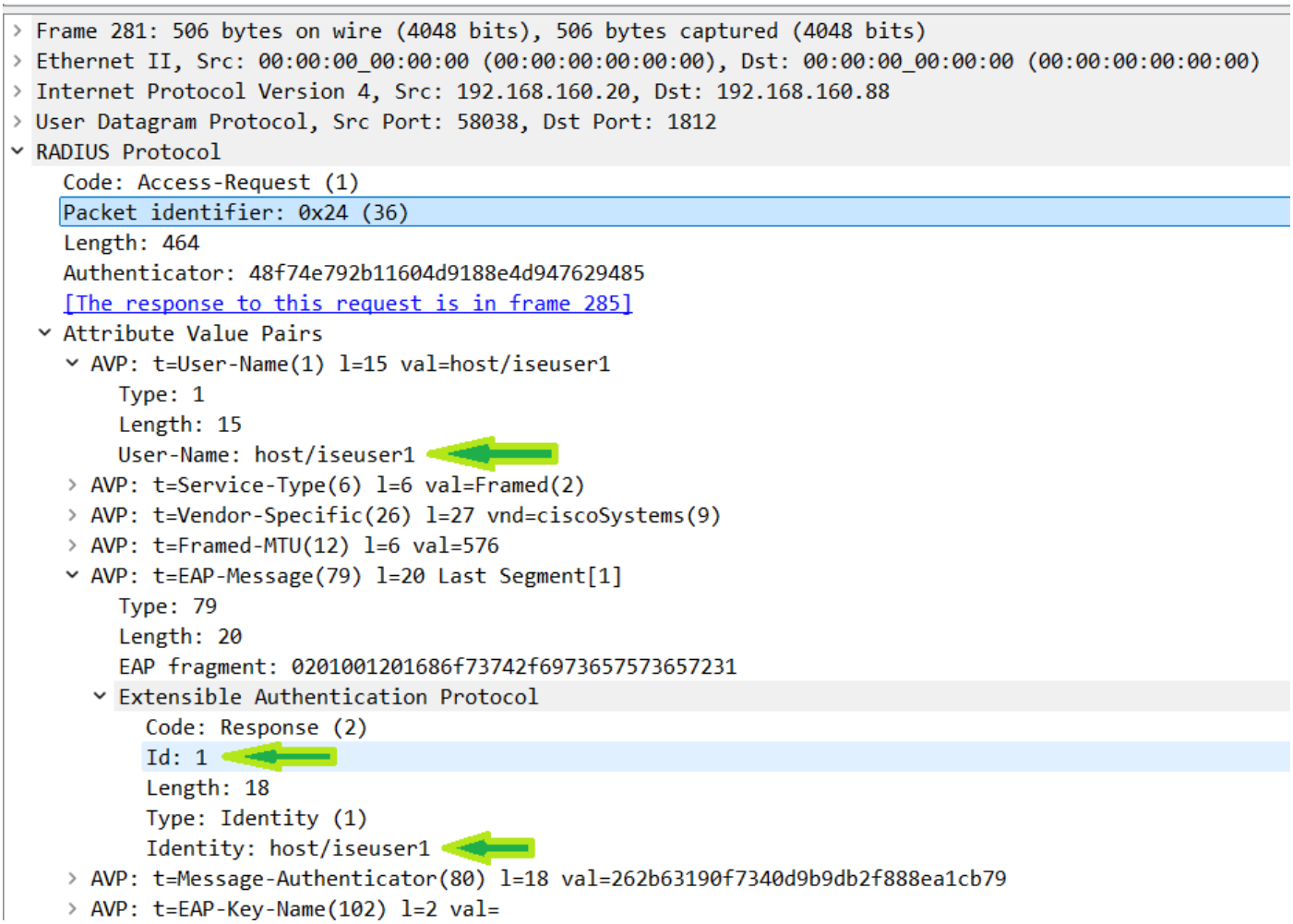
Access-RequestおよびAccess-Challenge

クライアントとの通信はEAPで、WLCとISE間の通信はRADIUSです。 RADIUS通信では、access-requestパケットとaccess-challengeパケットが使用されます。WLCはサプリカントからEAPパケットを受信し、RADIUSアクセス要求を使用してISEに転送します。稼働中のネットワークでは、ISEはaccess-challengeで応答します。

アクセス要求

これで、WLCはクライアントのIDを認識したため、そのクライアントがネットワークで許可されているかどうかをRADIUSサーバに確認する必要があります。そのために、WLCはaccess-requestパケットを送信して、そのクライアントのアクセスを要求します。WLCがEAPデータとともに送信するデータの他の部分があります。 これらをまとめて、発言者に応じて属性値ペア、AVP、またはAVペアと呼びます。

AVPについては、この説明の範囲外であるため、このドキュメントでは詳しく説明しません。ここでは、ユーザ名（EAPデータ）が含まれ、RADIUSサーバ（この場合はISE）に送信されたことを確認するだけです。また、EAP-ID番号1もISEに送信されていることがわかります。EAPパケットIDを地上波で表示したときも1だったことを思い出してください。ここで注意する最後の重要な点は、WLCがこれらのAVPをすべて追加したため、クライアントが送信した114バイトのパケットは、ISEに送信される前に488バイトのパケットに変換されることです。

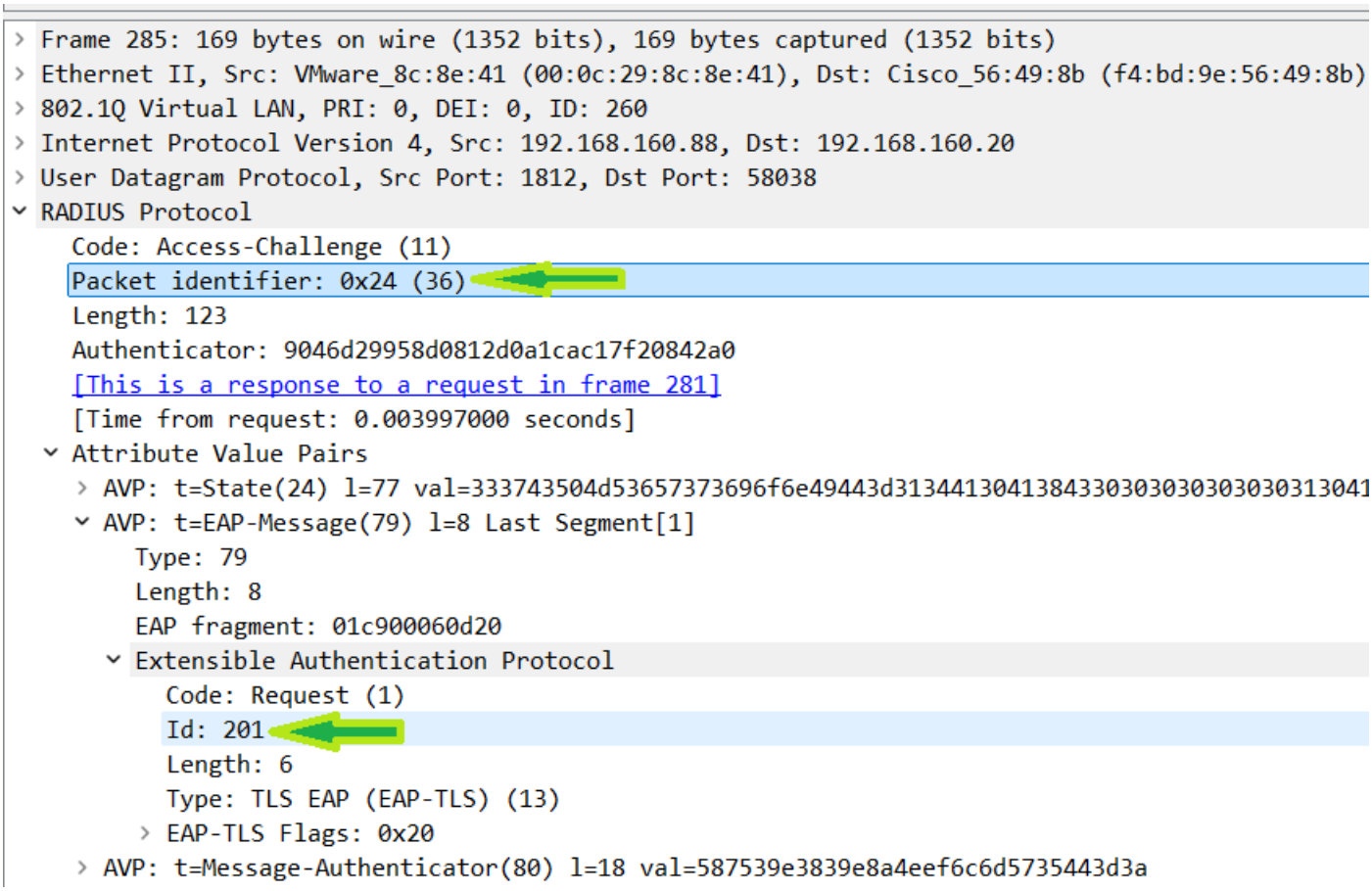


アクセスチャレンジ

ISEがアクセス要求を受信し、応答することを決定したと仮定すると、この応答はISEからのアクセスチャレンジとして到達することが期待されます。アクセス要求を振り返ると、AVPが起動する前にRADIUSパケットIDが36であることがわかります。

WLCがaccess-challengeを受信すると、RADIUS IDはaccess-requestのパケットIDと一致する必要があります。RADIUSパケットIDは、ISEとWLC間のRADIUS通信用です。また、このパケットでは、ISEがISEとクライアント間の通信の追跡に使用される新しいEAP IDとして201を設定していることがわかります。この時点で、WLCはISEとクライアント間の通信のパススルーにすぎません。

これらのパケットIDをすべてメモしておくことが重要です。そうすることで、圧縮のフローと、ネットワークでこれらのパケットを追跡する方法を理解できます。実稼働環境では、通常、同時に複数の認証が実行されます。calling-station-idコマンドを使用して、パケットとクライアントのMACアドレスを照合します。次に、RADIUSパケットIDとEAPパケットIDを使用して、この特定のクライアントのパケットフローを追跡できます。この時点では、どちらの側も証明書を送信していないため、MTUを気にする必要はありません。



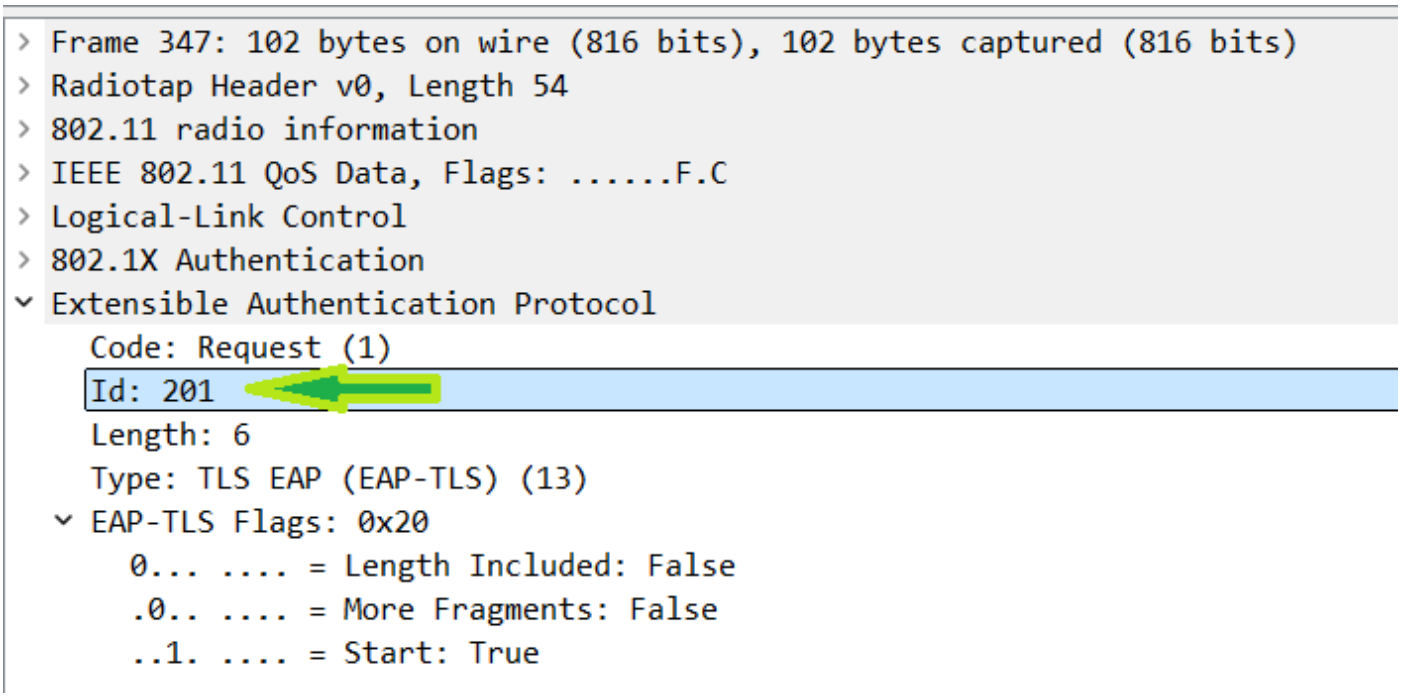
EAP要求とEAP応答

クライアントはRADIUSではなくEAPを使用します。つまり、WLCはアクセスチャレンジを受信すると、RADIUSデータを削除し、クライアントに送信できるようにEAP要求を取り出す必要があります。

EAP要求

これは、WLCがアクセスチャレンジを受信した際にアクセスチャレンジ内で行ったときとまったく同じでなければなりません。ただし、RADIUSの内容はすべて削除され、EAP部分だけがクライアントに送信されます。

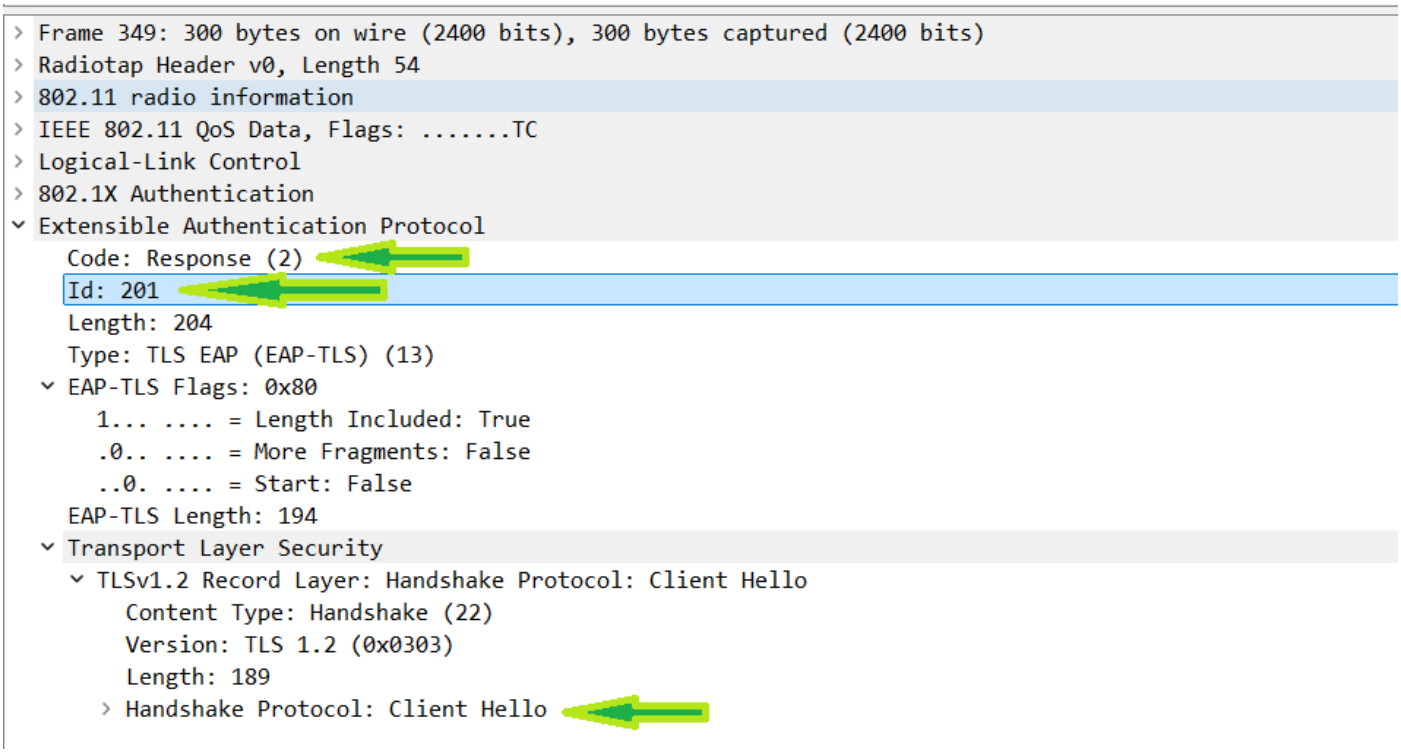
EAPパケットID 201は、WLCがISEから受信したデータと同じであるため、access-challengeのときと同じように表示されます。このフローはEAP-IDと同じですが、WLCから送信されず、EAP方式の確立に使用されます。このパケットは、EAP-TLSセッションの開始を確立するためだけなので、まだかなり小さいです。



EAP応答

クライアントはEAP-Requestを受信すると、EAP-Responseで応答する必要があります。 EAP-Responseで、クライアントがTLSセッションの確立を開始します。これは、TLSが使用される他の状況と同じように見えます。これは「client hello」メッセージで始まります。このドキュメントでは、client helloの内容については詳しく説明しません。このトピックとは無関係であるためです。ここで注意する必要があるのは、TLSセッションが設定されているということです。

他のTLS設定と同様に、パケット内のデータを確認できます。EAP-ID応答の場合と同様に、このパケットはWLCに到達し、アクセス要求に変換されます。ISEは、アクセスチャレンジにパッケージされたEAP要求で応答します。これからもこれがフローです。



TLS証明書

ここで、パケットサイズの増加が見られます。証明書は、1つ以上の中間認証局(CA)の存在に応じてかなり大きくなる可能性があります。 自己署名証明書の場合は、2つの中間CAとルートCAにチェーンされたデバイス証明書を持つ証明書よりも明らかにサイズが小さくなります。どちらの方法でも、通常は証明書の所有者が自身のパケットをここでフラグメント化し始めます。

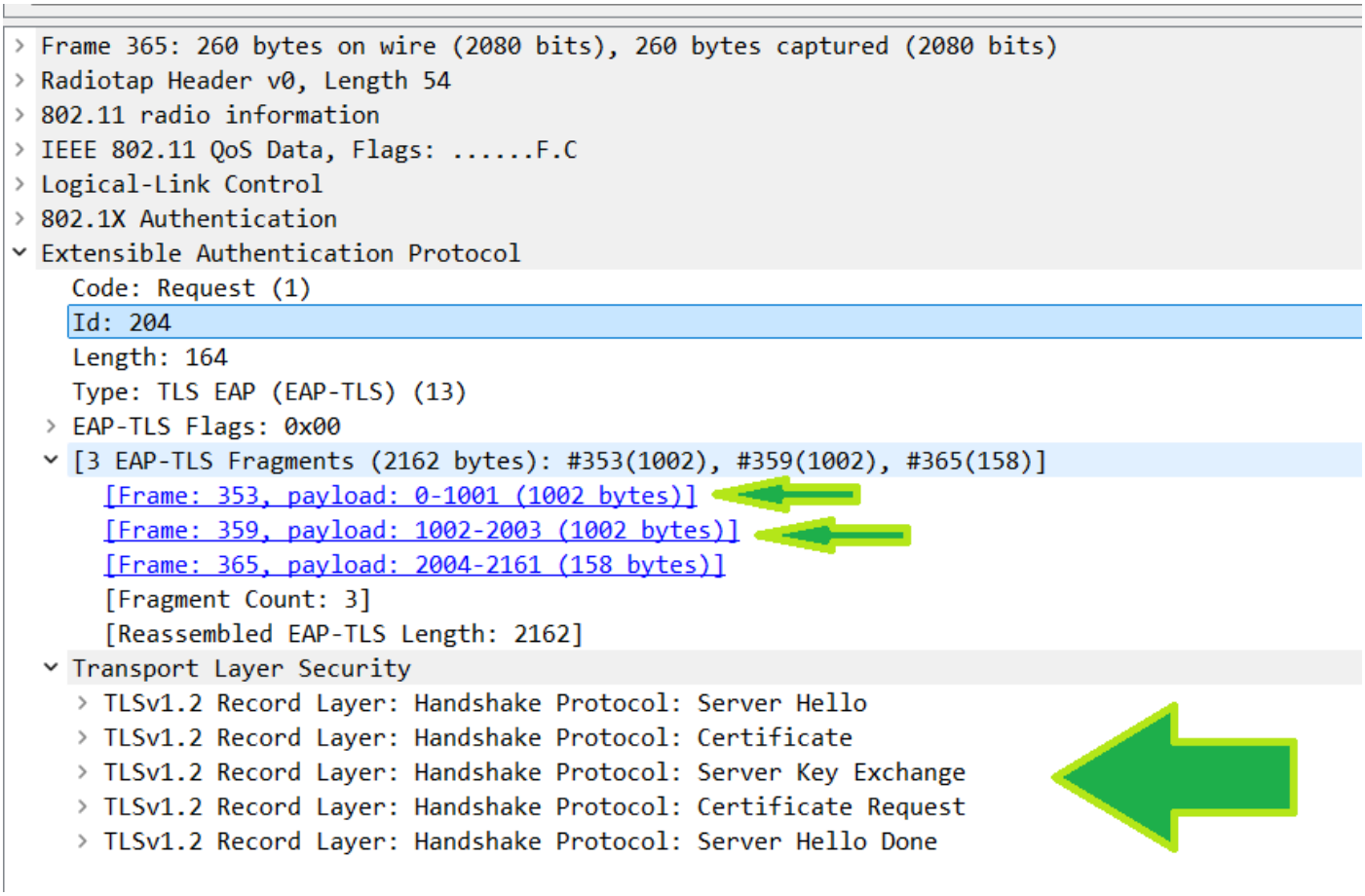
ISE証明書

ISEはTLSクライアントhelloを受信したので、別のEAP要求で応答します。この新しいEAP要求では、ISEは「server hello」メッセージ、その証明書、「server key exchange」、「certificate request」、および「server hello done」メッセージをすべて一度に送信します。これらすべてを1つのパケットで送信した場合、パケットはネットワークのMTUを超えることになります。そのため、ISEはパケット自体をフラグメント化して、MTUの範囲内に収めます。ISEでは、二重フラグメンテーションを回避するために、パケットのデータ部分が1002バイトを超えないようにフラグメント化されます。

二重フラグメンテーションとは何ですか。最初のフラグメンテーションはISEで発生します。これは、送信するデータがネットワークのMTU内に収まるには大きすぎるためです。それでも、MTUが同じであるにもかかわらず、ネットワークのセットアップ方法によっては、ネットワーク内の他の場所でパケットをフラグメント化して、ヘッダーを追加し、MTUの範囲内に留まるようにする必要がある場合があります。これは、do not fragmentビットがチェックされている場合でもtrueになる可能性があります。

VPNトンネルや、この問題に関連する任意のトンネルを使用する場合は、この例が適しています。データをVPNトンネルに送信するには、VPNルータがトラフィックにヘッダーを追加する必要があります。このRADIUSトラフィックがMTUまたはその近くでフラグメント化された場合、このVPNではMTUの下にデータを保持し、追加のヘッダーを追加する方法はありません。これは、後で説明するCAPWAPトンネルにも当てはまります。

そのため、別のデバイスで再びパケットがフラグメント化されるような状況に入るのを避けるために、ISEは、ほとんどのネットワークにおいてこれが回避できる場所でパケットをフラグメント化します。これは、ISEがこのデータを複数のEAP要求で送信し、毎回、空のEAP応答を待機することを意味します。EAP IDは、フラグメントが送信されるたびに増加します。WLCの観点からは、すべてのフラグメントのアクセスチャレンジおよびアクセス要求交換が行われ、フラグメントが送信されるたびにRADIUSパケットIDが増加します。



クライアント証明書

ISEがすべてのフラグメントを送信し、それらがクライアントによって再構成されると、パケットフローは何かを送信するためにクライアントに移ります。TLSでは、認証を完了するために、この時点でクライアントが自身の証明書を送信することが想定されます。ここで物事が複雑になります。ISEと同様に、クライアントは複数のTLSパーツを一度に送信します。そのうちの1つが証明書です。

ISEで確認されたものとは異なり、ほとんどのクライアントはMTUのすぐ下でEAPデータを送信します。このデモでは、802.1xデータは1492です。これに関する問題は、APがWLCに送信できるようにCAPWAPヘッダーを追加する必要があることです。

どうすればよいでしょうか。APは、ヘッダーを追加してWLCに送信できるように、パケットをフラグメント化する必要があります。APは、フラグメント化せずにWLCにパケットを取得することはできません。とは言うものの、ここではパケットが二重にフラグメント化され、最初にクライアントから、次にAPから送信されます。ただし、このフラグメンテーションはCAPWAPで予期されるため、通常は問題にはなりません。

パケットの地上波：

```
> 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
```

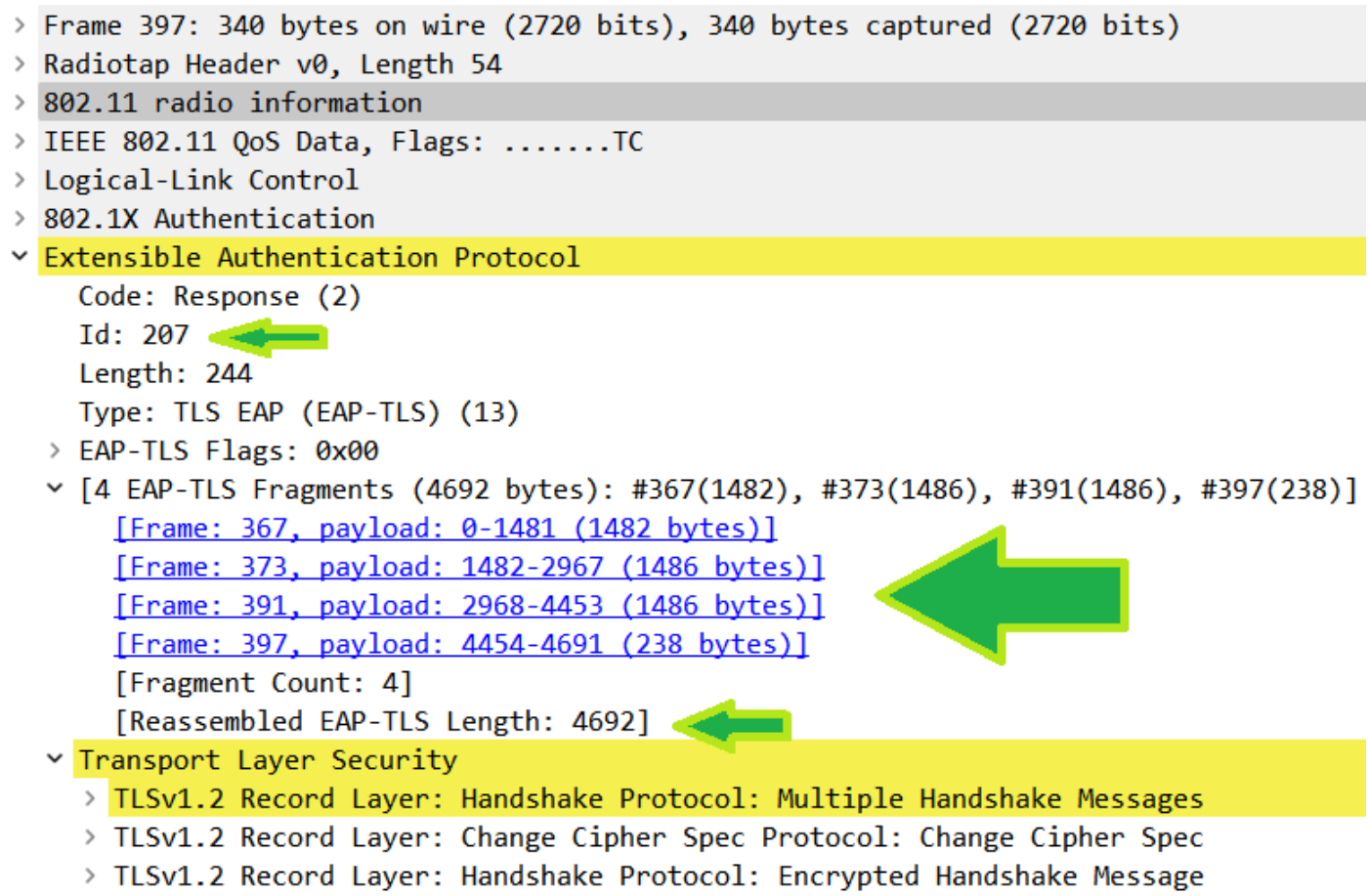
回線上のパケットフラグメント：

```
> 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]
```

回線上で再構成されたパケット：

```
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
```

すべてのクライアントフラグメントが空中で再構成されました。



WLCでのクライアント証明書

WLCは2つのCAPWAPフラグメントを受信し、これらを再構成してクライアントからの1492バイトのパケット全体を取得し、パケットを復元します。ただし、長くは復元されません。WLCがアクセス要求を送信する方法を振り返ると、ISEにデータを送信する前にパケットに約400バイト相当のRADIUS AVPを追加する必要があることを思い出す必要があるため、この復元は短期間しか続きません。

単純な計算として、WLCが408バイトを追加して合計パケットサイズを1900にするとします。これは1500 MTUをはるかに超えているので、WLCは何を行いますか。パケットを再度フラグメント化します。

この時点で、WLCはデフォルトで1396でパケットをフラグメント化します。ここで考えることは、ISEの場合と同じです。パケットが別のトンネルを通過する必要がある場合に、ヘッダーを追加するために再度フラグメント化する必要がないように、パケットを十分に小さくすることが望まれます。ただし、WLCはISEほど慎重ではないため、ここでは1396で十分です。

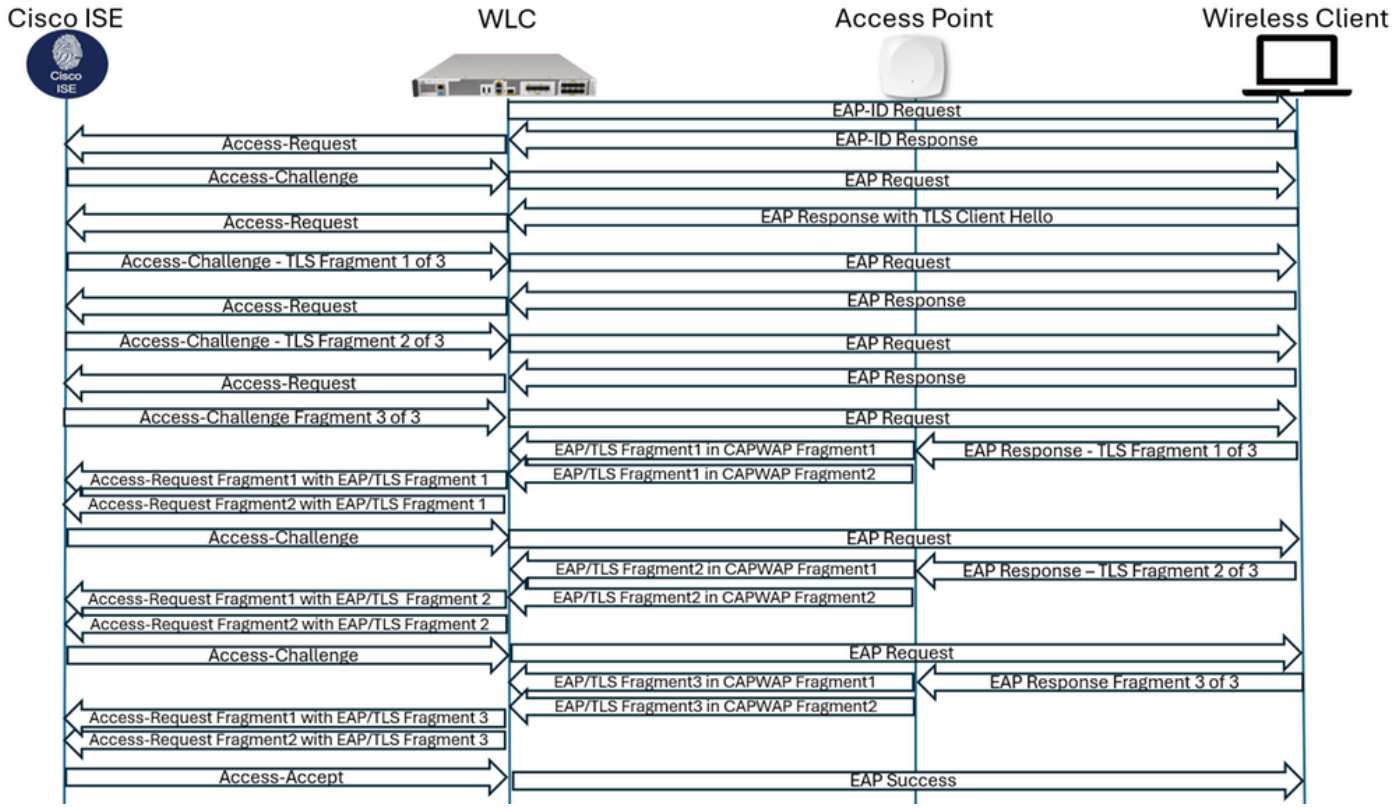
WLCから送信される断片化パケット：

```
> Frame 318: 1414 bytes (11312 bits), 1414 bytes captured (11312 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 (1376 bytes)
    Data: e2b6071407f152b7012807e9e3a7b0f3ca162bfd8d2c29b6eaae21a7010f686f73742f69…
    [Length: 1376]
```

```
> Frame 319: 695 bytes (5560 bits), 695 bytes captured (5560 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
> User Datagram Protocol, Src Port: 58038, Dst Port: 1812
v RADIUS Protocol
    Code: Access-Request (1)
    Packet identifier: 0x28 (40)
    Length: 2025
    Authenticator: e3a7b0f3ca162bfd8d2c29b6eaae21a7
    [The response to this request is in frame 322]
  v Attribute Value Pairs
    > AVP: t=User-Name(1) l=15 val=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
    > AVP: t=EAP-Message(79) l=255 Segment[1]
    > AVP: t=EAP-Message(79) l=255 Segment[2]
    > AVP: t=EAP-Message(79) l=255 Segment[3]
    > AVP: t=EAP-Message(79) l=255 Segment[4]
    > AVP: t=EAP-Message(79) l=255 Segment[5]
    v AVP: t=EAP-Message(79) l=229 Last Segment[6]
        Type: 79
        Length: 229
        EAP fragment: 8bc4be38a7487cb8dcaf6e1664bb495f72cf96e0c91b6c40c64ec67de3fcdaf15cb73989…
      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
    > AVP: t=Message-Authenticator(80) l=18 val=ffcd8b97d2d366fd9d995043bfe27607
    > AVP: t=EAP-Key-Name(102) l=2 val=
    > AVP: t=Vendor-Specific(26) l=49 vnd=ciscoSystems(9)
```

パケットフローTL;DR

ISEは証明書を送信するときに、1002バイトのTLSパケットをフラグメント化します。問題はありません。クライアントが自身の証明書を送信する場合、通常はフラグメント化はMTUに近づきます。APはパケットにCAPWAPヘッダーを追加する必要があるため、このパケットもフラグメント化する必要があります。WLcがフラグメントを受信したら、パケットを再構成する必要がありますが、RADIUS AVPを追加して、パケットが再度フラグメント化されるようにする必要があります。 パケットフローは次のようになります。



RADIUS MTU動作の変更

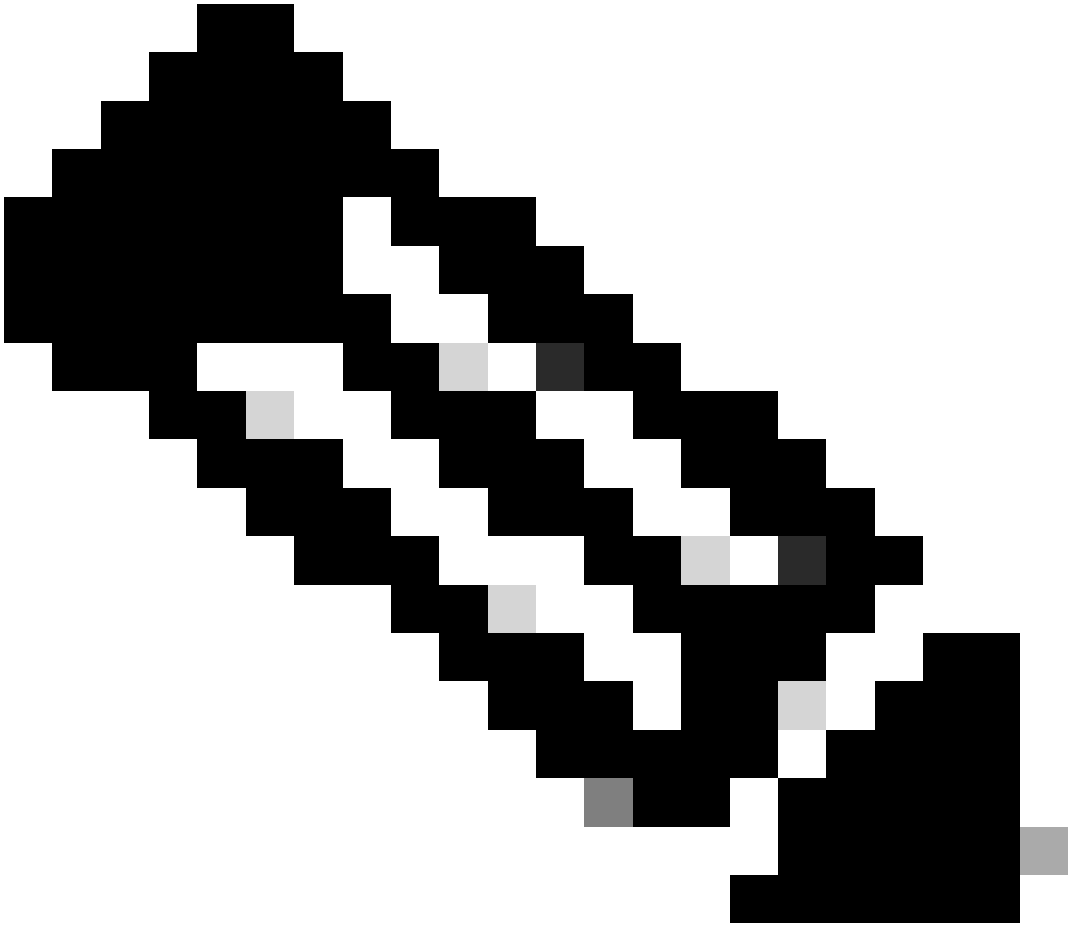
ワイヤレスクライアントのデータトラフィックのパケットフローを調べてみると、ワイヤレスインフラストラクチャが影響を与えている場所は数ヵ所しかありません。最初の場所はトラフィックがAPから発信される場所で、2番目の場所はトラフィックがWLCから発信される場所です。例外は、ワイヤレスインフラストラクチャがクライアントMSSを調整できるTCPトラフィックです。ただし、EAPはTCPには該当せず、実際には独自のプロトコルです。

EAPとRADIUSのトラフィックフローを見ると、元のパケットサイズがMTUに近づきすぎた場合に、ネットワークがAPとWLCの両方のトラフィックサイズに実際に影響を与えていることがわかります。この交換におけるWLCの役割を適切に理解すると、RADIUSパケットのサイズの影響を受ける場所が実際には1ヵ所だけであることがわかります。これは、EAP応答が着信し、それをRADIUSアクセス要求に変更した場合です。

変更内容

EAP応答がMTUを超えている場合は、RADIUS AVPを追加した後、フラグメント化する必要があります。このパケットは何があってもフラグメント化する必要があるため、少なくともどのサイズでフラグメント化するかを決定できます。 そこで、17.11で導入された動作変更が登場します。

Cisco Bug ID CSCwc81849で追跡される機能要求で、RADIUSジャンボパケットのサポートを追加したいと考えています。この方法では、RADIUSパケットは1396で自動的にフラグメント化されなくなりました。ここで、ip radius source-interface <interface X>コマンドを追加すると、RADIUS access-requestはそのインターフェイスのMTUで送信されます。



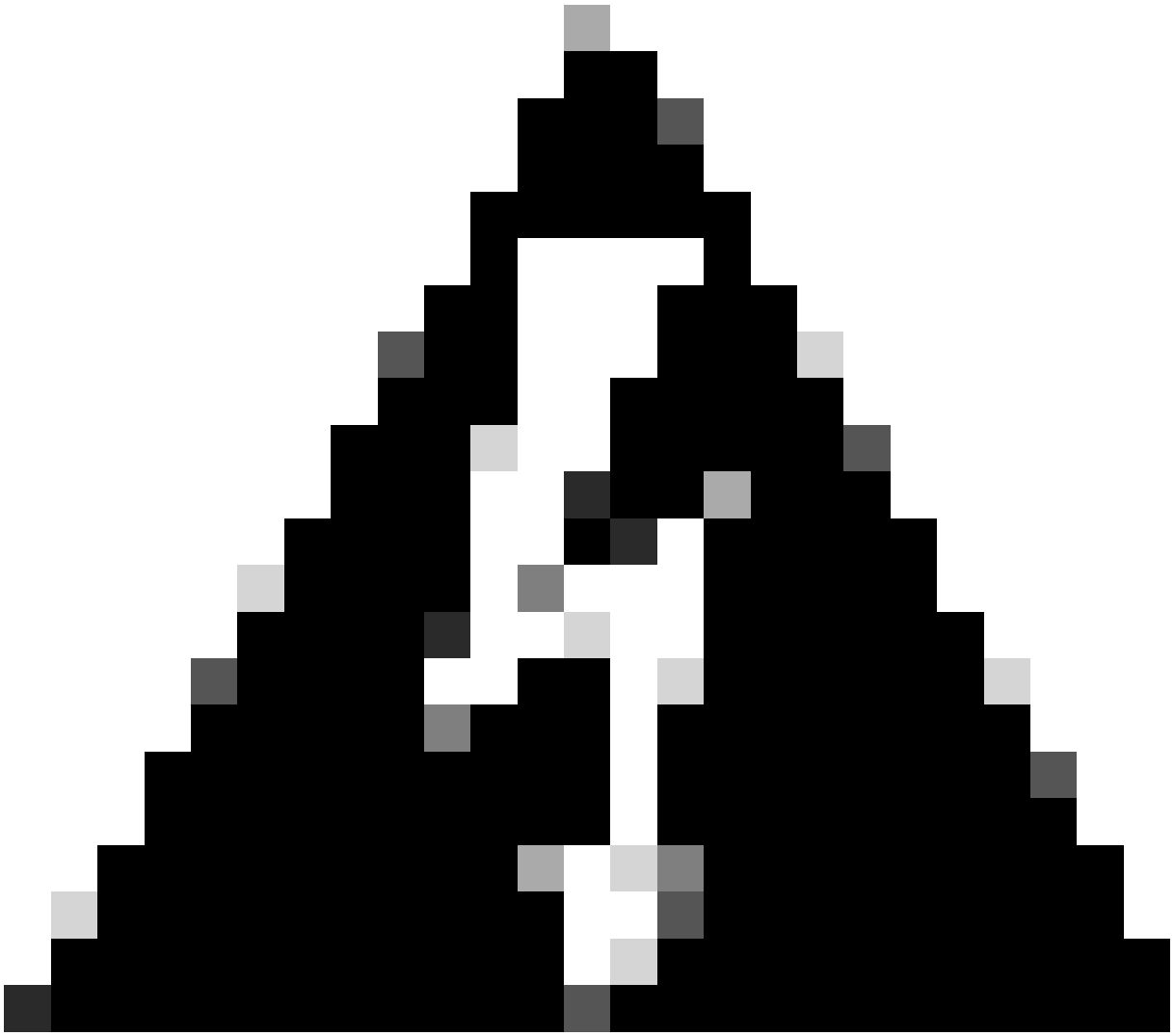
注:Cisco Catalyst Centerを使用している場合、AAA設定をプロビジョニングすると、送信元インターフェイスがサーバグループに自動的に追加されます。これにより、そのコマンドで使用されるインターフェイスのMTUサイズでフラグメント化するようにデフォルトの動作が変更されます。

この変更の使用方法

すべてのインターフェイスのデフォルトMTUは1500であるため、フラグメンテーションを行う新しいMTUになります。すべてのRADIUSトラフィックに使用されるデフォルトのインターフェイスは、ワイヤレス管理インターフェイス(WMI)です。 サーバグループの設定を確認すると、source-interfaceが指定されていない場合、WLCはWMIを使用してRADIUSトラフィックを1396で送信します。ただし、サーバグループの設定に移動し、source-interfaceがWMIであると指定した場合、WLCはWMIを使用したままRADIUSトラフィックを1500で送信します。

ここで、前述のVPNのようなデバイスがネットワーク内にあると仮定します。トラフィックを二重にフラグメント化することは望ましくありません。そのため、パケットを別の場所でフラグメント化するために、インターフェイスのMTUを小さい値に変更できます。MTUを1200などに変更

すると、パケットが1500バイトではなく1200バイトのマークでフラグメント化されます。



警告: WMIのMTUを変更すると、WLC管理IPアドレスとの間でやり取りされるすべてのトラフィックに影響します。

WMIのMTUを変更したくない場合でも、発信元インターフェイスを指定する目的は、WMIから別のインターフェイスに変更し、そのインターフェイスをRADIUSトラフィックに使用すること、およびそのインターフェイスのMTUを変更することです。この設定はサーバグループレベルで行われるため、この変更がどのRADIUSトラフィックに影響を与えるかを明確に指定できます。

この設定は、AAAサーバやWLANに関連付けられていません。同じサーバを含む複数のサーバグループを設定し、必要に応じてそのうちの1つだけにソースインターフェイスを指定することもできます。このサーバグループは、方式リストに追加されてから、WLANに追加されます。そのため、たとえば、この変更を行うWLANが1つしかない場合、AAAサーバが1つしかない場合でも、新しいサーバグループを作成し、使用するMTUのインターフェイスを指すip radius source-interface コマンドを使用して、この新しいグループにAAAサーバを追加し、この新しいグループを使用して新しい方式リストを作成し、この変更を行う特定のWLANに方式リストを追加できま

す。

警告：稼働中のネットワークに変更を加える際には、常にメンテナンス時間帯に行うことをお勧めします。

証明はパケットキャプチャにあります

一般的に知られているnネットワーキングでは、キャプチャしなければ証明できないいくつかの設定例と、その動作を示すこれらの変更を示します。

次にWLAN設定を示します。テスト中は、方式リストで使用されているサーバグループのみが変更されます。

```
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
!
```

デフォルトのMTUを使用したSource-Interfaceコマンドの追加

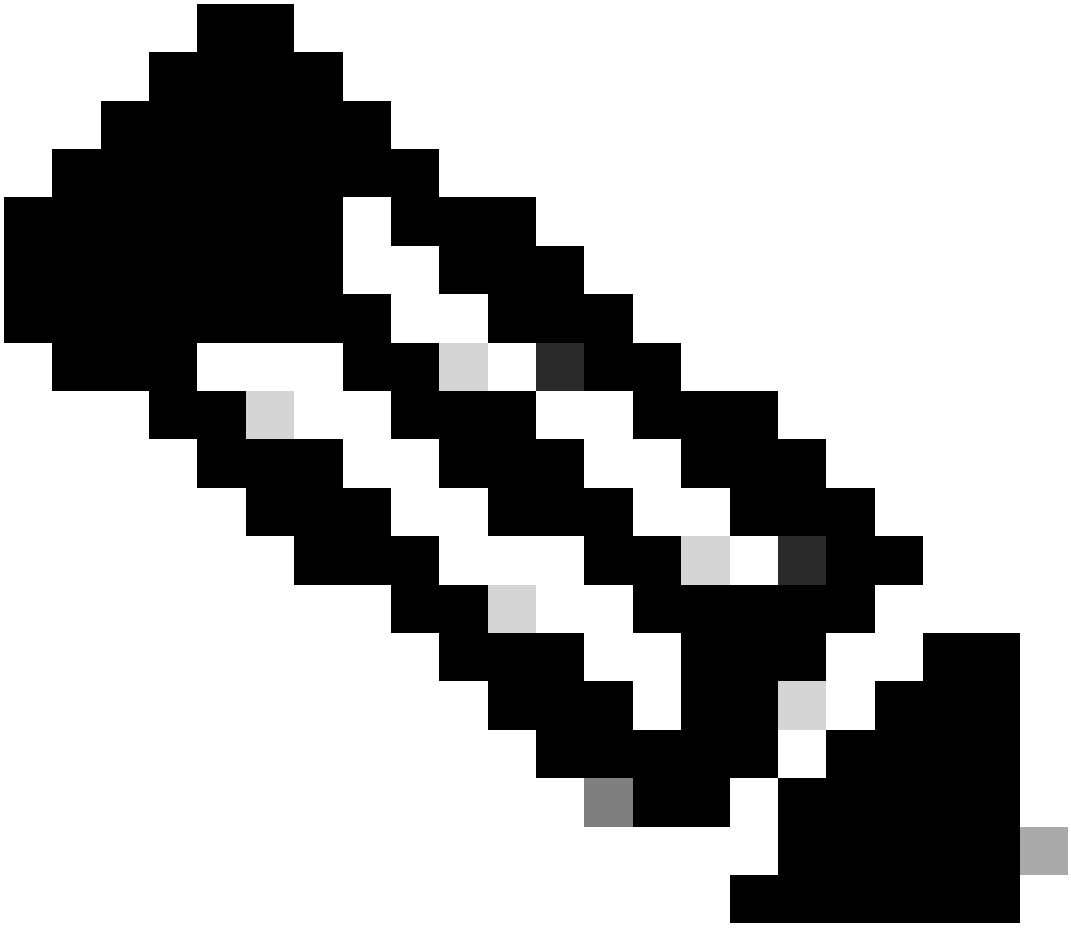
ここでは、ISEサーバをポイントする通常のサーバグループです。MTUが設定されていないWMIを示すsource interfaceコマンドが追加されました。設定は次のようになります。

```
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
```

WLANに関連付けられている認証方式リストにNoMTUサーバグループが追加されていることがわかります。このサーバグループにはip radius source-interface VLAN260コマンドが使用されており、VLAN 260ではMTUを指定しません。つまり、1500のMTUを使用しています。1500というMTUがあることをご確認いただくだけで、show run allコマンドを使用して、出力中のインターフェイスを探すことができます。

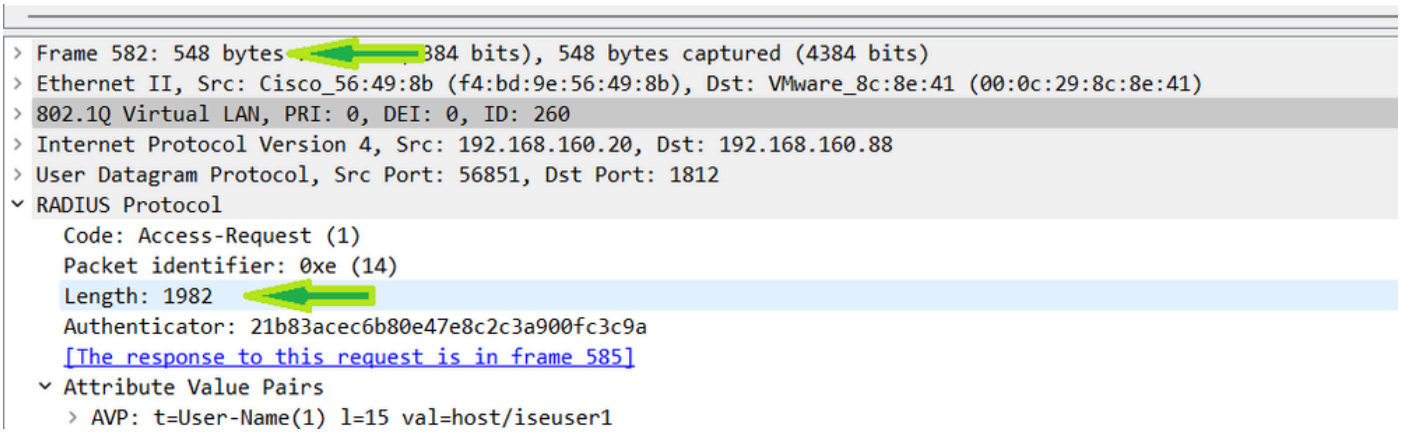
```
interface Vlan260
 ip address 192.168.160.20 255.255.255.0
 no ip clear-dont-fragment
 ip redirects
 ip unreachables
 no ip proxy-arp
 ip mtu 1500
```

次に、WLCがRADIUSデータを追加した後、クライアント証明書をISEに送信する必要があるパケットを調べます。



注：この場合、行のバイト数は1518です。これには、VLANヘッダーやレイヤ2ヘッダーなど、イーサネットペイロードの外部のヘッダーが含まれます。これらはMTUにはカウントされません。

```
> 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]
```



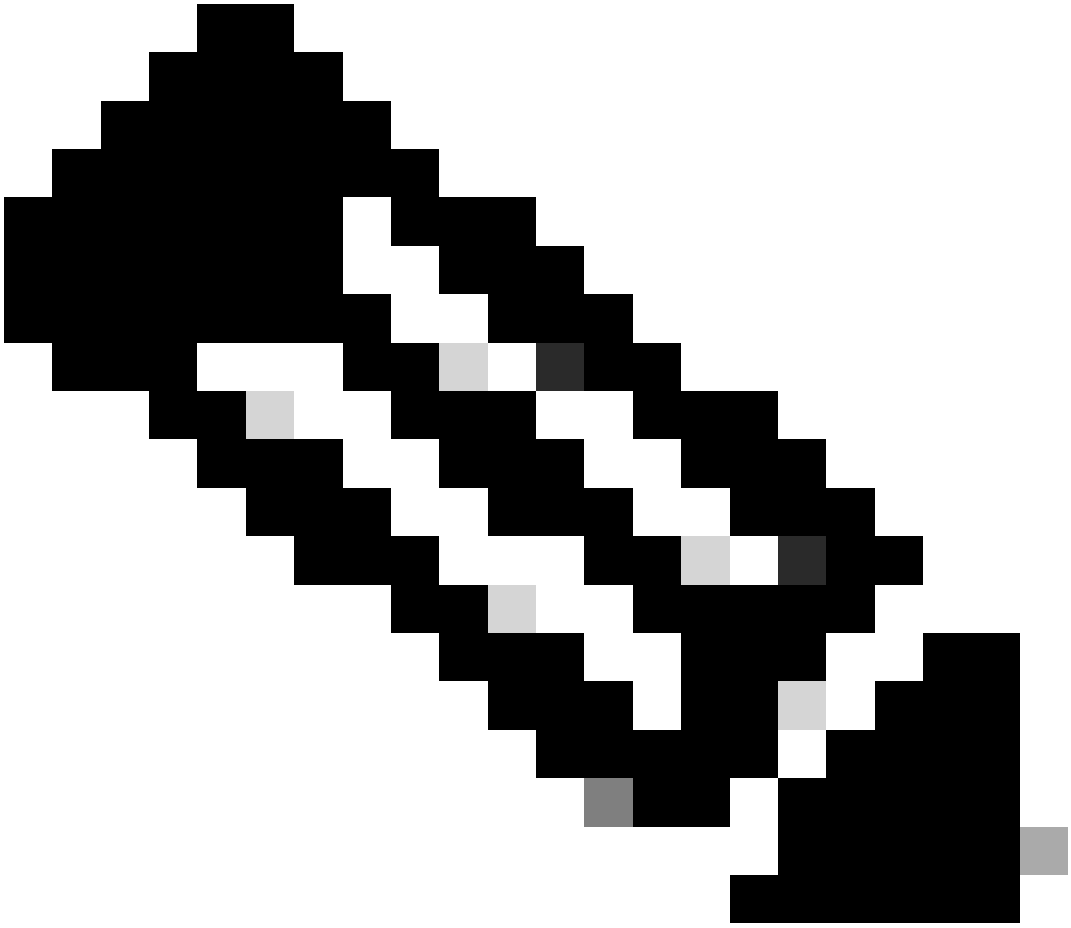
ここでは、データ部分が1480でフラグメント化されていることがわかります。このフラグメントは、WMIで1500 MTU未満で取得できます。次のパケットは550バイト未満ですが、RADIUSデータの合計サイズは1982であることがわかります。ただし、新しいMTUでフラグメント化すれば正常に動作します。

Mtuが1200の非WMIインターフェイスを使用する

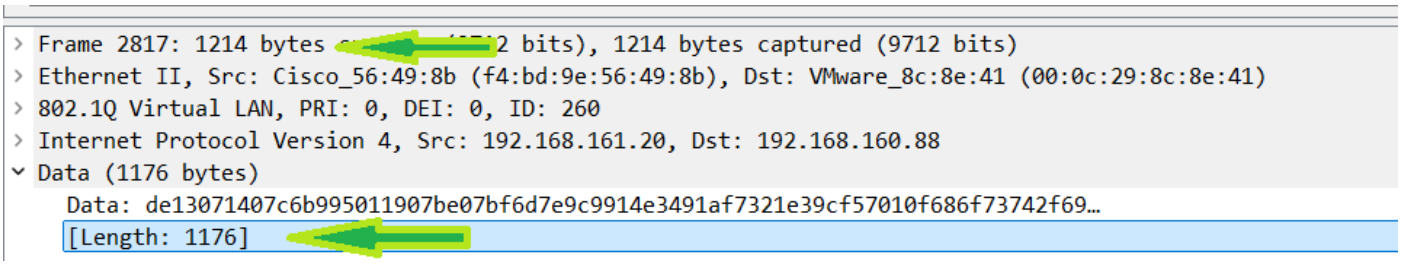
ここで、より小さいMTUでフラグメント化し、この変更が他のトラフィックに影響を与えないようにするとします。ここでは問題はありません。ソースインターフェイス設定がこの目的のために作成されたSVIを指すだけなので、設定は同じままです。この新しいサーバグループを指すように方式リストを変更します。このサーバグループは、私のWMIではなく、MTUが1200に設定されている発信元インターフェイスを使用します。設定は次のようになります。

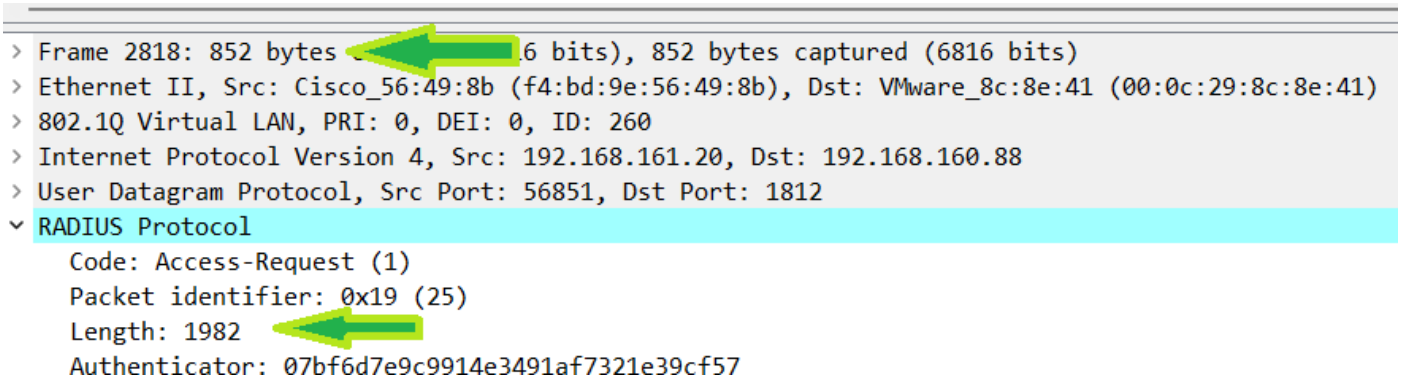
```
9800#show run aaa
!
aaa authentication dot1x TLS-AuthC group MTU1200
!
!
radius server ISE
 address ipv4 192.168.160.10 auth-port 1812 acct-port 1813
 key 6 _`gINMNXObFibbBMhNMFAbKUAAB
!
aaa group server radius MTU1200
 server name ISE
 ip radius source-interface Vlan261
 deadtime 5
!
9800#show run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

次に、この低いMTUでパケットがどのように見えるかを確認します。



注:MTUを小さくしてフラグメンテーションポイントを変更することは、新しい動作の一部ではありません。これはいつも本当でした。1396でフラグメント化するデフォルトの動作がMTUに収まらない場合は、常に異なるポイントでフラグメント化します。このセクションでは、使用可能なオプションについて説明します。





ここでは、RADIUSデータはまだ1982バイトですが、今回はデータがデフォルトの1376ではなく1176でフラグメント化されました。発信元インターフェイスが使用されていなければ、このデータはフラグメント化されます。MTUを1500に設定してsource-interfaceコマンドを使用すると、1480でフラグメント化されることに注意してください。ここで設定を使用すると、WLC上の他のトラフィックに干渉することなく、トラフィックをより低いMTUに操作できます。

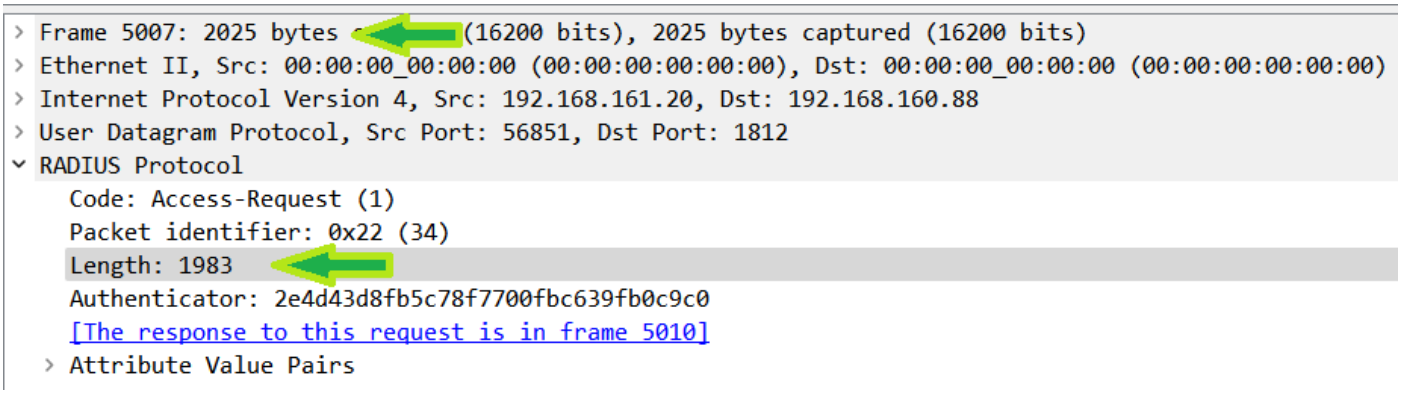
ジャンボフレームに9000のMTUを使用

ジャンボフレームを送信するオプションの機能が作成されたため、VLAN 261の非WMIインターフェイスを使用して同様のテストを行わないのは残念です。ただし、この時点でIP MTUは9000に設定されています。簡単に説明すると、SVIでIP MTUを設定できるようにするには、IP MTUよりも大きな値にMTUを設定する必要があります。これは次の設定で確認できます。

```
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
```

キャプチャを見ると、パケットがフラグメント化されていないことがわかります。これは、1983でRADIUSデータサイズを持つ1つのパケットとして送信されました。これが機能するためには、このサイズのパケットが通過できるようにネットワークの他の部分を設定する必要があることに注意してください。

ここで注意すべきもう1つの点は、クライアントのMTUが変更されていないため、クライアントはEAPパケットを1492でフラグメント化していることです。違いは、WLCはクライアントデータをフラグメント化することなく、パケットをISEに送信するために必要なすべてのRADIUSデータを追加できることです。



結論

EAP-TLSを使用する場合、クライアントは証明書をAAAサーバに送信する必要があります。通常、これらの証明書はMTUよりも大きいため、クライアントは証明書をフラグメント化する必要があります。クライアントがデータをフラグメント化するポイントは、MTUに非常に近いです。APはCAPWAPヘッダーを追加する必要があるため、クライアントが送信している内容をフラグメント化する必要があります。WLCはこの2つのパケットを受信し、それらを元に戻しますが、RADIUSデータを追加するために再度フラグメント化する必要があります。この時点で、ネットワーク管理者は、クライアントが送信したEAPパケットをWLCでフラグメント化する方法を制御できます。

ip radius source-interface <interface you want to use>コマンドをAAAサーバグループに追加すると、WMIの代わりに（またはWMIを含めて）、WLCは配置したインターフェイスを使用します。このコマンドを使用すると、WLCに対して、デフォルトの1396ではなく、そのインターフェイスのMTUを使用してフラグメント化するように指示されます。これにより、パケットがネットワーク内を移動する方法をより細かく制御できます。

Cisco Catalyst Centerを使用する場合、source interfaceコマンドがサーバグループに追加され、デフォルトの動作が変更されます。

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。