Title

Process for identifying WLAN 802.1X security settings through protocol analysis.

Brief

As part of a previous role I assessed existing WLAN’s for Voice client support.

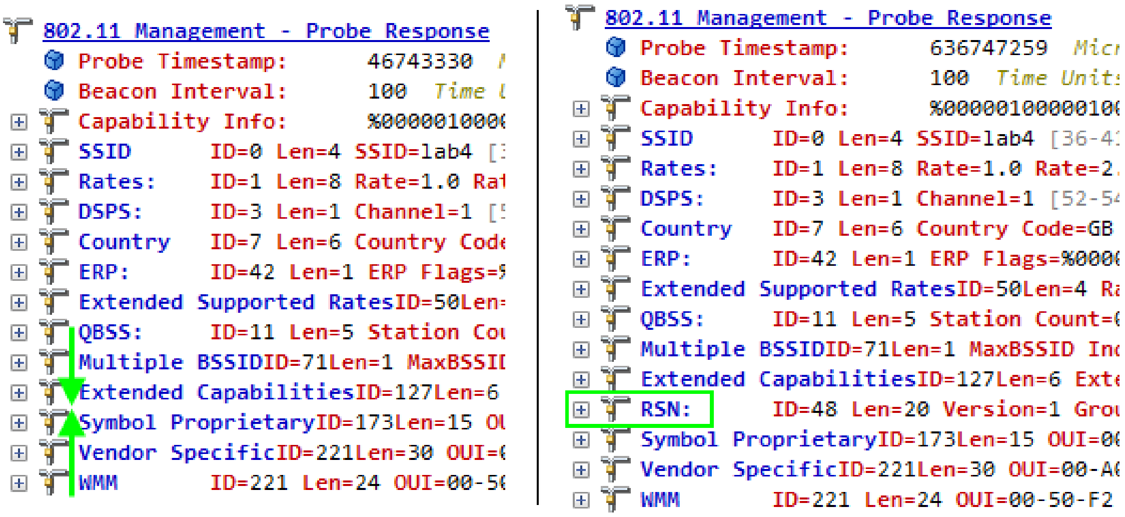
During these assessments it was necessary for me to verify the WLAN setup met the requirements of our voice client, including details such as channels used, data rates supported, and security employed. Often and understandably customers like the DoD would refuse for security reasons to provide a configuration output from their equipment. However, experience has taught me that you cannot always rely on the details a customer provides you, and verifying with your own tools is prudent.

A lot of configuration details are easily discovered from the Beacon and Probe Response frames for a WLAN. One that is not so easy to verify is the EAP type in use. Below is an internal document I wrote to educate my peers on 802.11 security and how they can use Protocol Analysis to verify the Dot1X security of a WLAN.

Diagnosis

Normally information about the security employed on a WLAN can be found in the RSN IE (Information Element) of Beacons and Probe Responses. RSN stands for Robust Security Network and was introduced in 802.11i to improve the security of 802.11 networks.

To illustrate this point Figure 1 shows a Probe Response from a WLAN using Open authentication (left) and a WLAN using WPA2 authentication (right). You can see the RSN IE (highlighted) in the WPA2 Probe Response below, but missing in the Open Authentication (pre-802.11i) Probe Response.



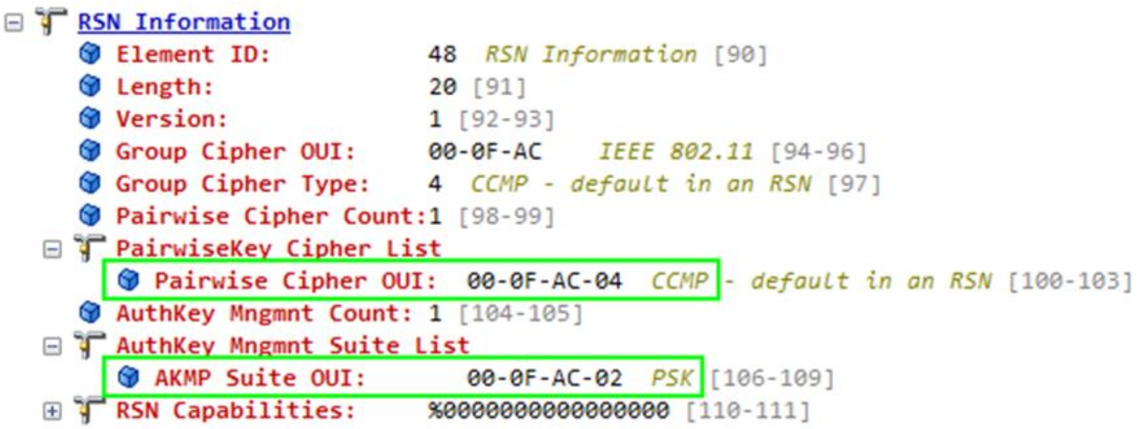
```
802.11 Management - Probe Response
   Probe Timestamp:        46743330  M
   Beacon Interval:        100   Time U
⊞  Capability Info:        %0000010000
⊞  SSID        ID=0 Len=4 SSID=lab4 [3
⊞  Rates:      ID=1 Len=8 Rate=1.0 Rat
⊞  DSPS:       ID=3 Len=1 Channel=1 [5
⊞  Country     ID=7 Len=6 Country Code
⊞  ERP:        ID=42 Len=1 ERP Flags=%
⊞  Extended Supported RatesID=50Len=
⊞  QBSS:       ID=11 Len=5 Station Cou
⊞  Multiple BSSIDID=71Len=1 MaxBSSID
⊞  Extended CapabilitiesID=127Len=6
⊞  Symbol ProprietaryID=173Len=15 OU
⊞  Vendor SpecificID=221Len=30 OUI=0
⊞  WMM         ID=221 Len=24 OUI=00-50
```

```
802.11 Management - Probe Response
   Probe Timestamp:        636747259  Micr
   Beacon Interval:        100   Time Units
⊞  Capability Info:        %00000100000100
⊞  SSID        ID=0 Len=4 SSID=lab4 [36-41
⊞  Rates:      ID=1 Len=8 Rate=1.0 Rate=2.
⊞  DSPS:       ID=3 Len=1 Channel=1 [52-54
⊞  Country     ID=7 Len=6 Country Code=GB
⊞  ERP:        ID=42 Len=1 ERP Flags=%0000
⊞  Extended Supported RatesID=50Len=4 Ra
⊞  QBSS:       ID=11 Len=5 Station Count=
⊞  Multiple BSSIDID=71Len=1 MaxBSSID Ind
⊞  Extended CapabilitiesID=127Len=6 Exte
⊞  RSN:        ID=48 Len=20 Version=1 Grou
⊞  Symbol ProprietaryID=173Len=15 OUI=00
⊞  Vendor SpecificID=221Len=30 OUI=00-A0
⊞  WMM         ID=221 Len=24 OUI=00-50-F2
```

Figure 1 - 802.11i RSN Information Element

WPA2 is the authentication and key management method for all 802.11i WLAN's, whether they use a Pre Shared Key or EAP (Extensible Authentication Protocol). Often people assume WPA2 is only used on WLAN's with a "password" (pre-shared key) and EAP is something different. This is not the case, WPA2 is used to define the secure handshake process for both PSK and EAP WLAN's, hence you will see in Microsoft Windows options for WPA2-Personal (PSK) and WPA2-Enterprise (EAP).

Figure 2 shows the RSN IE of a WPA2-PSK Probe Response. Without the magic of Protocol Analysis software we would have to know that 00-0F-AC-04 meant CCMP was in use for encryption and that 00-0F-AC-02 meant that a Pre Shared Key (PSK) was in use for access credentials. Instead, Omnipeek puts a lovely mucus green note on the end telling us this (Wireshark does this also in a less revolting colour). CCMP is the 802.11i derivative of AES used to encrypt WPA2 connections. When you see CCMP you can read it as AES.

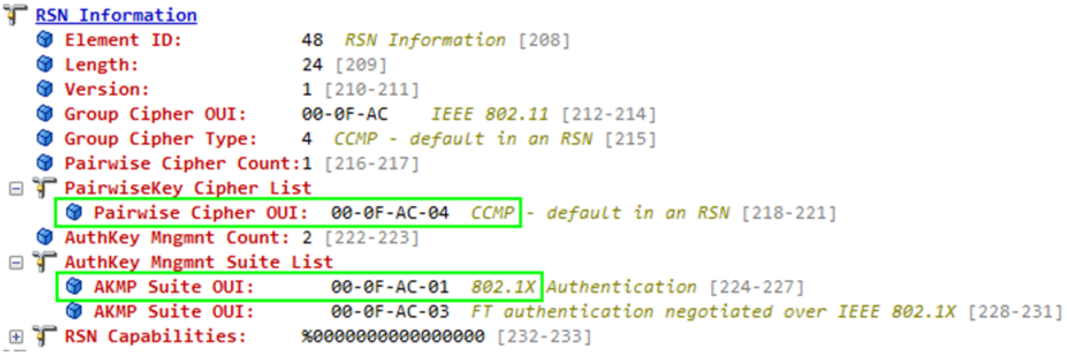


```
⊟ RSN Information
    Element ID:             48  RSN Information [90]
    Length:                 20 [91]
    Version:                1 [92-93]
    Group Cipher OUI:       00-0F-AC    IEEE 802.11 [94-96]
    Group Cipher Type:      4  CCMP - default in an RSN [97]
    Pairwise Cipher Count:1 [98-99]
  ⊟ PairwiseKey Cipher List
      Pairwise Cipher OUI:   00-0F-AC-04  CCMP - default in an RSN [100-103]
    AuthKey Mngmnt Count: 1 [104-105]
  ⊟ AuthKey Mngmnt Suite List
      AKMP Suite OUI:        00-0F-AC-02  PSK [106-109]
  ⊞ RSN Capabilities:       %0000000000000000 [110-111]
```

Figure 2 - RSN using PSK

When it comes to EAP security the 802.11 standard supports the EAP *framework* rather than any single EAP mechanism. For some reason EAP has become the terminology used for RADIUS based credential access on 802.11 networks. However, it is actually EAP *encapsulated* in the IEEE 802.1X standard (or Dot1X as its commonly known) that is used to secure our “EAP” WLAN’s.

Figure 3 shows a Probe Response from a WPA2-EAP WLAN. The AKMP Suite is listed as 802.1X, not EAP and you won’t find the EAP type (PEAP, EAP-TLS, etc) listed in any Beacons or Probe Responses. This is because the AP is going to use 802.1X between the client and itself. The encapsulated EAP frames will then be sent onto the Authenticating Server (typically RADIUS server) who decides which EAP type to use. In fact, when configuring a WLAN for EAP you won’t typically be able to specify an EAP type to use (unless you’re using the controller/access point as the Authenticating Server as well as the Authenticator).

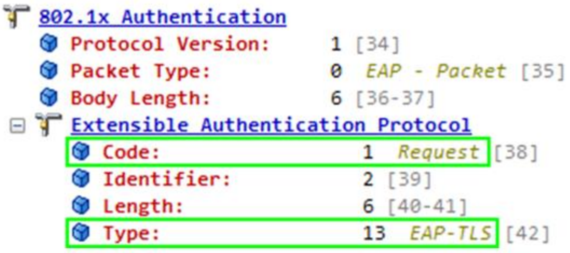


```
RSN Information
    Element ID:             48  RSN Information [208]
    Length:                 24 [209]
    Version:                1 [210-211]
    Group Cipher OUI:       00-0F-AC    IEEE 802.11 [212-214]
    Group Cipher Type:      4  CCMP - default in an RSN [215]
    Pairwise Cipher Count:1 [216-217]
⊟ PairwiseKey Cipher List
      Pairwise Cipher OUI:   00-0F-AC-04  CCMP - default in an RSN [218-221]
    AuthKey Mngmnt Count: 2 [222-223]
⊟ AuthKey Mngmnt Suite List
      AKMP Suite OUI:        00-0F-AC-01  802.1X Authentication [224-227]
      AKMP Suite OUI:        00-0F-AC-03  FT authentication negotiated over IEEE 802.1X [228-231]
⊞ RSN Capabilities:       %0000000000000000 [232-233]
```

Figure 3 - RSN using EAP

Note: The above Probe Response shows the WLAN also supports 802.11r or Fast Transition (FT) as it’s known in the standard.

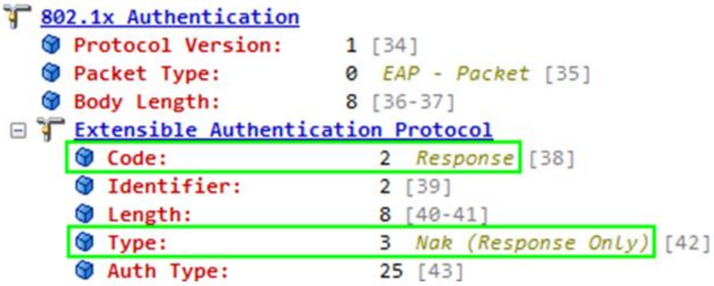
You are able to use the Beacons and Probe Responses to verify that the WLAN is using WPA2-EAP but to verify the EAP variety supported by the Authenticating Server a connection must be attempted and the EAP messages (known as EAP Over LAN or EAPOL messages) reviewed. Figure 4 shows the Authenticating Server asking a Supplicant (in this case an iPhone) to use EAP-TLS for authentication.



```
802.1x Authentication
    Protocol Version:         1 [34]
    Packet Type:              0  EAP - Packet [35]
    Body Length:              6 [36-37]
⊟ Extensible Authentication Protocol
        Code:                     1  Request [38]
        Identifier:               2 [39]
        Length:                   6 [40-41]
        Type:                     13  EAP-TLS [42]
```

Figure 4 - EAPOL TLS Request

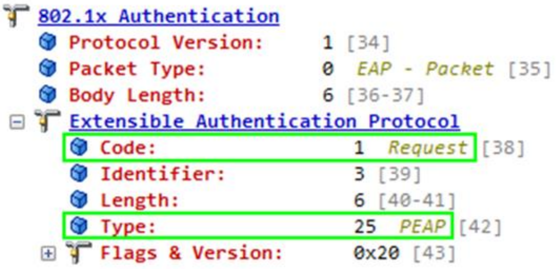
To support EAP-TLS a certificate needs loaded onto the client. As this process is aimed at exploring the security supported by the WLAN (rather than successfully connecting to it) and every customer visited uses different client certificates this is deliberately skipped. Therefore, the Supplicant sends a Negative Acknowledgement (Nak) to the Authenticating server declining EAP-TLS, as can be seen in Figure 5. These frames all happen in sequence, so you can tell the Nak is for the previous EAP-TLS request.



```
802.1x Authentication
    Protocol Version:         1 [34]
    Packet Type:              0  EAP - Packet [35]
    Body Length:              8 [36-37]
⊟ Extensible Authentication Protocol
        Code:                     2  Response [38]
        Identifier:               2 [39]
        Length:                   8 [40-41]
        Type:                     3  Nak (Response Only) [42]
        Auth Type:                25 [43]
```

Figure 5 - EAPOL TLS Negative-Ack

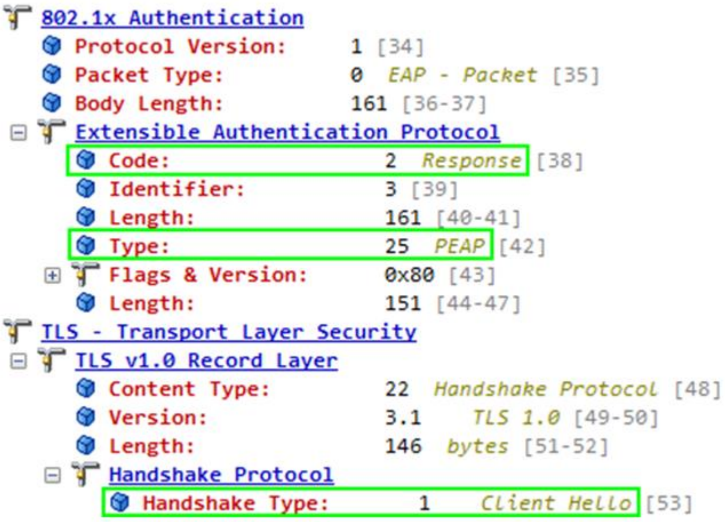
In this example the Authenticating Server was setup supporting multiple authentication mechanisms, so it now requests the Supplicant uses PEAP for authentication as shown in Figure 6.



```
802.1x Authentication
    Protocol Version:         1 [34]
    Packet Type:              0  EAP - Packet [35]
    Body Length:              6 [36-37]
⊟ Extensible Authentication Protocol
        Code:                     1  Request [38]
        Identifier:               3 [39]
        Length:                   6 [40-41]
        Type:                     25  PEAP [42]
    ⊞ Flags & Version:           0x20 [43]
```

Figure 6 - EAPOL PEAP Request

This time the Supplicant does support PEAP so it Responds reiterating the use of PEAP, and starts the RADIUS handshake process with a “Hello”, as seen in Figure 7.



```
802.1x Authentication
    Protocol Version:       1 [34]
    Packet Type:            0  EAP - Packet [35]
    Body Length:            161 [36-37]
  ⊟ Extensible Authentication Protocol
      Code:                   2  Response [38]
      Identifier:             3 [39]
      Length:                 161 [40-41]
      Type:                   25  PEAP [42]
    ⊞ Flags & Version:        0x80 [43]
      Length:                 151 [44-47]
TLS - Transport Layer Security
  ⊟ TLS v1.0 Record Layer
      Content Type:           22  Handshake Protocol [48]
      Version:                3.1    TLS 1.0 [49-50]
      Length:                 146  bytes [51-52]
    ⊟ Handshake Protocol
        Handshake Type:          1    Client Hello [53]
```

Figure 7 - EAPOL PEAP Response

There will then be lots more EAPOL messages to follow as the Supplicant and Authenticating Server continue the challenge and keying involved in the EAP mechanism in use.

All of this information is contained in the 802.11 headers so is viewable without needing to decrypt any captures.

Summary

Wi-Fi Scanners such as WiFi Explorer and WinFi do not list the EAP type supported by a WLAN as it is not declared in any management frames. One of the quickest independent ways to verify this is to capture an association attempt and explorer the negotiation between the Authenticating Server and Supplicant.

I hope this essay demonstrates my understanding of the security mechanisms supported by the 802.11 standard and the tools used to implement and troubleshoot them.