

# 802.11 WPA2-Enterprise/EAP/dot1x over-the-airワイヤレススニファを復号化するためのWiresharkとFreeRADIUSの設定

## 内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[手順](#)

[ステップ1:Access-accept PacketからPMKを復号化します。](#)

[ステップ2:PMKを抽出します。](#)

[ステップ3:OTAスニファを復号化します。](#)

[復号化された802.11パケットの例](#)

[暗号化された802.11パケットの例](#)

[関連情報](#)

## 概要

このドキュメントでは、Extensible Authentication Protocol(EAP)方式を使用して、Wi-Fi Protected Access 2(WPA2-Enterprise)または802.1x(dot1x)暗号化ワイヤレスover-the-air(OTA)スニファを復号化する方法について説明します。

完全な4方向EAP over LAN(EAPoL)ハンドシェイクがキャプチャされている限り、PSKベース/WPA2パーソナル802.11 OTAキャプチャを比較的簡単に復号化できます。ただし、事前共有キー(PSK)は、セキュリティの観点から必ずしも推奨されるわけではありません。ハードコードされたパスワードを解読するのは時間の問題です。

そのため、多くの企業では、無線ネットワークの優れたセキュリティソリューションとしてRemote Authentication Dial-In User Service(RADIUS)を使用してdot1xを選択しています。

## 前提条件

### 要件

次の項目に関する知識があることが推奨されます。

- radsniffがインストールされたFreeRADIUS
- Wireshark/Omnipeek、または802.11ワイヤレストラフィックを復号化できる任意のソフトウェア
- ネットワークアクセスサーバ(NAS)とオーセンティケーター間の共有秘密を取得する権限を取得する権限を付与します。

- EAPセッション全体で、最初のアクセス要求 ( NASからオーセンティケータへ ) から最後のアクセス許可 ( オーセンティケータからNASへ ) まで、NASとオーセンティケータ間のRADIUSパケットキャプチャをキャプチャする機能
- 4方向EAPoLハンドシェイクを含むOver-the-Air(OTA)キャプチャを実行する機能

## 使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

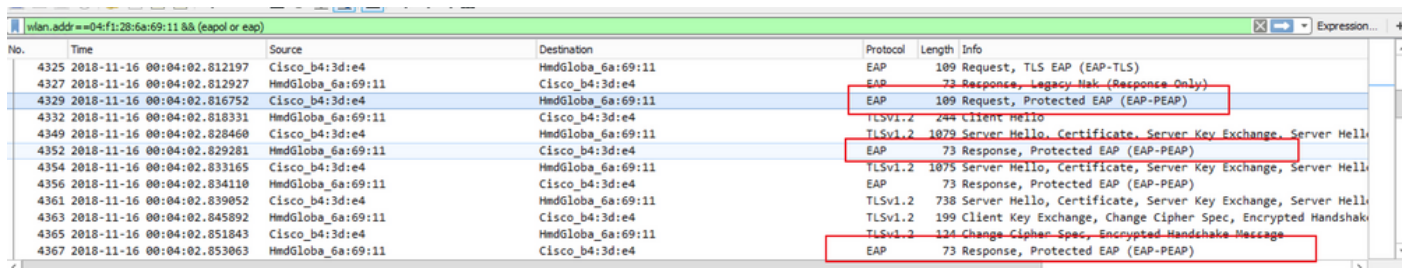
- Radiusサーバ ( FreeRADIUSまたはISE )
- Over-the-Airキャプチャデバイス
- Apple macOS/OS XまたはLinuxデバイス

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、初期 ( デフォルト ) 設定の状態から起動しています。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。

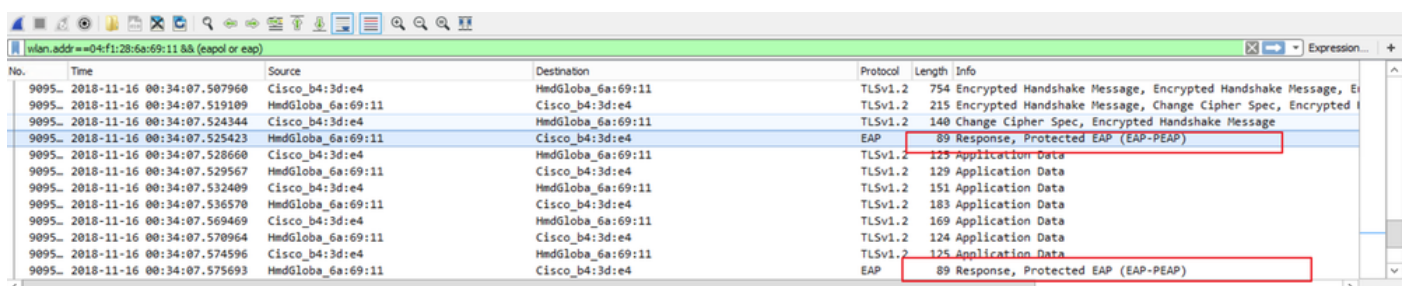
## 背景説明

この例では、2つのペアワイズマスターキー(PMK)は、ISE 2.3からキャプチャされたRADIUSパケットから取得されます。このSSIDのセッションタイムアウトは1800秒で、ここで指定されるキャプチャの長さは34分 ( 2040秒 ) です。

図に示すように、例としてEAP-PEAPが使用されますが、これは任意のdot1xベースのワイヤレス認証に適用できます。



No.	Time	Source	Destination	Protocol	Length	Info
4325	2018-11-16 00:04:02.812197	Cisco_b4:3d:e4	HmdGloba_6a:69:11	EAP	109	Request, TLS EAP (EAP-TLS)
4327	2018-11-16 00:04:02.812927	HmdGloba_6a:69:11	Cisco_b4:3d:e4	EAP	73	Response, Legacy Nak (Response Only)
4329	2018-11-16 00:04:02.816752	Cisco_b4:3d:e4	HmdGloba_6a:69:11	EAP	109	Request, Protected EAP (EAP-PEAP)
4332	2018-11-16 00:04:02.818331	HmdGloba_6a:69:11	Cisco_b4:3d:e4	TLSv1.2	244	Client Hello
4349	2018-11-16 00:04:02.828460	Cisco_b4:3d:e4	HmdGloba_6a:69:11	TLSv1.2	1075	Server Hello, Certificate, Server Key Exchange, Server Hello
4352	2018-11-16 00:04:02.829281	HmdGloba_6a:69:11	Cisco_b4:3d:e4	EAP	73	Response, Protected EAP (EAP-PEAP)
4354	2018-11-16 00:04:02.833165	Cisco_b4:3d:e4	HmdGloba_6a:69:11	TLSv1.2	1075	Server Hello, Certificate, Server Key Exchange, Server Hello
4356	2018-11-16 00:04:02.834110	HmdGloba_6a:69:11	Cisco_b4:3d:e4	EAP	73	Response, Protected EAP (EAP-PEAP)
4361	2018-11-16 00:04:02.839052	Cisco_b4:3d:e4	HmdGloba_6a:69:11	TLSv1.2	738	Server Hello, Certificate, Server Key Exchange, Server Hello
4363	2018-11-16 00:04:02.845892	HmdGloba_6a:69:11	Cisco_b4:3d:e4	TLSv1.2	199	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
4365	2018-11-16 00:04:02.851843	Cisco_b4:3d:e4	HmdGloba_6a:69:11	TLSv1.2	124	Change Cipher Spec, Encrypted Handshake Message
4367	2018-11-16 00:04:02.853063	HmdGloba_6a:69:11	Cisco_b4:3d:e4	EAP	73	Response, Protected EAP (EAP-PEAP)



No.	Time	Source	Destination	Protocol	Length	Info
9095_	2018-11-16 00:34:07.507960	Cisco_b4:3d:e4	HmdGloba_6a:69:11	TLSv1.2	754	Encrypted Handshake Message, Encrypted Handshake Message, Encrypted Handshake Message
9095_	2018-11-16 00:34:07.519109	HmdGloba_6a:69:11	Cisco_b4:3d:e4	TLSv1.2	215	Encrypted Handshake Message, Change Cipher Spec, Encrypted Handshake Message
9095_	2018-11-16 00:34:07.524344	Cisco_b4:3d:e4	HmdGloba_6a:69:11	TLSv1.2	140	Change Cipher Spec, Encrypted Handshake Message
9095_	2018-11-16 00:34:07.525423	HmdGloba_6a:69:11	Cisco_b4:3d:e4	EAP	89	Response, Protected EAP (EAP-PEAP)
9095_	2018-11-16 00:34:07.528660	Cisco_b4:3d:e4	HmdGloba_6a:69:11	TLSv1.2	125	Application Data
9095_	2018-11-16 00:34:07.529567	HmdGloba_6a:69:11	Cisco_b4:3d:e4	TLSv1.2	129	Application Data
9095_	2018-11-16 00:34:07.532409	Cisco_b4:3d:e4	HmdGloba_6a:69:11	TLSv1.2	151	Application Data
9095_	2018-11-16 00:34:07.536570	HmdGloba_6a:69:11	Cisco_b4:3d:e4	TLSv1.2	183	Application Data
9095_	2018-11-16 00:34:07.569469	Cisco_b4:3d:e4	HmdGloba_6a:69:11	TLSv1.2	169	Application Data
9095_	2018-11-16 00:34:07.570964	HmdGloba_6a:69:11	Cisco_b4:3d:e4	TLSv1.2	124	Application Data
9095_	2018-11-16 00:34:07.574596	Cisco_b4:3d:e4	HmdGloba_6a:69:11	TLSv1.2	125	Application Data
9095_	2018-11-16 00:34:07.575693	HmdGloba_6a:69:11	Cisco_b4:3d:e4	EAP	89	Response, Protected EAP (EAP-PEAP)

## 手順

### ステップ1: Access-accept PacketからPMKを復号化します。

PMKを抽出するには、NASとオーセンティケータ間のRADIUSキャプチャに対してradsniffを実行します。キャプチャ中に2つのaccess-acceptパケットが抽出される理由は、この特定のSSIDでセッションタイムアウトタイマーが30分に設定され、キャプチャの長さが34分であることです。認



としてカウントできます。ただし、ログに示されたこの状態でradsniffがスタックしている場合は、同じNASとオーセンティケータの間で別の長いパケットキャプチャ(B)を使用して、このパケットキャプチャ(A)をカスケードしてください。次に、カスケードされたパケット(A+B)に対してradsniffコマンドを実行します。パケットキャプチャ(B)の唯一の要件は、radsniffコマンドを実行して詳細な結果を表示できることです。


```
FRLU-M-51X5:pcaps frlu$ radsniff -I /Users/frlu/Downloads/radius_novlan.pcap -s Cisco123 -x
```

```
Logging all events
```

```
Sniffing on (/Users/frlu/Downloads/radius_novlan.pcap)
```

この例では、WLCパケットロギング機能によってキャプチャされたワイヤレスLANコントローラ(WLC)コントロールプレーンロギング(A)が、ISEのTCPダンプ(B)からより長いキャプチャにカスケードされています。WLCパケットロギングは通常、非常に小さいサイズであるため、例として使用されます。

### WLCパケットロギング(A)

 radius_novlan.pcap	Pcap N...apture	22 KB	Today at 11:56 am
--	-----------------	-------	-------------------

### ISE Tcpdump(B)

 radius_eap_decode_Cisco123.pcap	Yesterday at 12:04 pm	850 KB	Pcap N...apture
---	-----------------------	--------	-----------------

### マージ(A+B)

 radius_novlan_merged.pcapng	Pcapn...Capture	927 KB	Today at 12:28 pm
---	-----------------	--------	-------------------

次に、マージされたpcap(A+B)に対してradsniffを実行すると、詳細な出力が表示されます。

```
FRLU-M-51X5:pcaps frlu$ radsniff -I /Users/frlu/Downloads/radius_novlan_merged.pcapng -s <shared-secret between NAS and Authenticator> -x
```

```
<snip>
```

```
2018-11-16 11:39:01.230000 (24) Access-Accept Id 172
/Users/frlu/Downloads/radius_novlan_merged.pcapng:10.66.79.42:32771 <- 10.66.79.36:1812 +0.000
+0.000
```

```
<snip>
```

### ステップ2:PMKを抽出します。

その後、詳細出力から各MS-MPPE-Recv-Keyの0xフィールドを削除し、ワイヤレストラフィックデコードに必要なPMKを提示します。

```
MS-MPPE-Recv-Key =
0xddb0b09a7d6980515825950b5929d02f236799f3e8a87f163c8ca41a066d8b3b
```

```
PMK:
ddb0b09a7d6980515825950b5929d02f236799f3e8a87f163c8ca41a066d8b3b
```

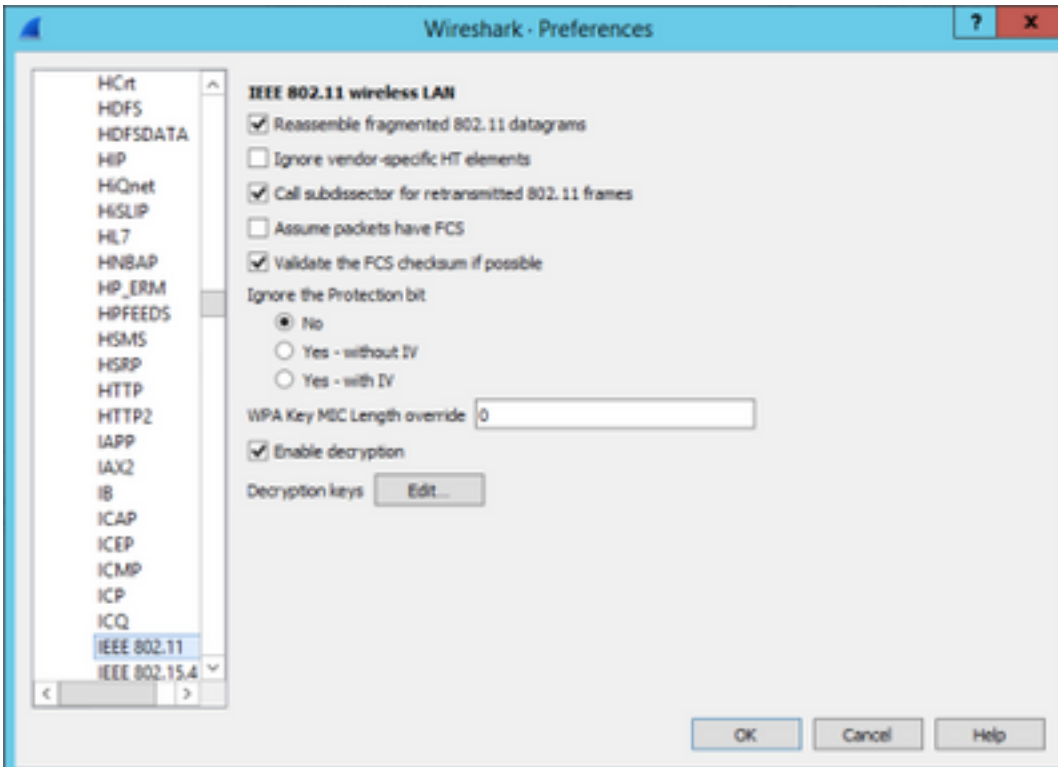
```
MS-MPPE-Recv-Key =
0x7cce47eb82f48d8c0a91089ef7168a9b45f3d798448816a3793c5a4dfb1cfb0e
```

PMK:

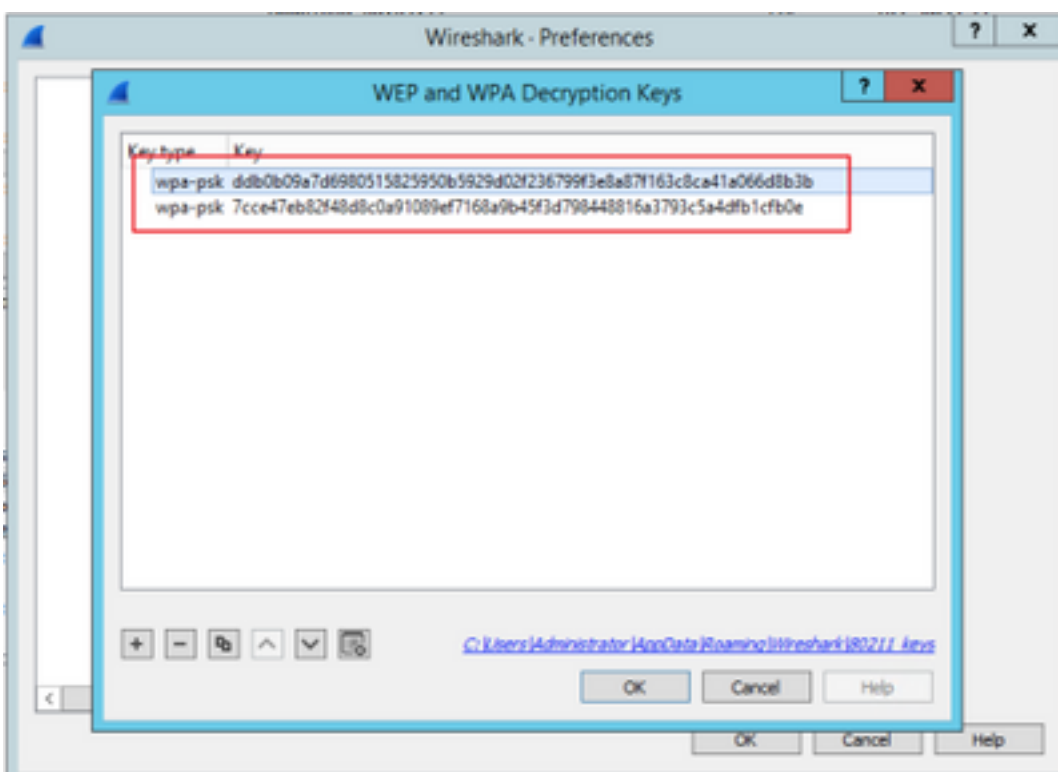
7cce47eb82f48d8c0a91089ef7168a9b45f3d798448816a3793c5a4dfb1cfb0e

### ステップ3:OTAスニファを復号化します。

[Wireshark] > [Preferences] > [Protocols] > [IEEE 802.11]に移動し、[Enable Decryption]をオンにし、[Decryption Keys]の横にある[Edit]ボタンをクリックします。



次に、キーの種類としてwpa-pskを選択し、[Key]フィールドに導出されたPMKを入力し、[OK]をクリックしてください。これが完了すると、OTAキャプチャが復号化され、上位レイヤ(3+)情報が表示されます。



## 復号化された802.11パケットの例

The screenshot shows a Wireshark capture of an 802.11 network. The packet list pane shows several packets, with packet 397886 highlighted in red. The packet details pane for packet 397886 shows the following structure:

- Frame 397886: 154 bytes on wire (1232 bits), 154 bytes captured (1232 bits)
- Radiotap Header v0, Length 48
- 802.11 radio information
- IEEE 802.11 QoS Data, Flags: .p....TC
- Logical-Link Control
- Internet Protocol Version 4, Src: 172.16.255.13, Dst: 40.127.66.24
- Transmission Control Protocol, Src Port: 45658, Dst Port: 80, Seq: 128, Ack: 4001196, Len: 0

The packet bytes pane shows the raw data of the packet, with the QoS data field highlighted in blue.

PMKが含まれていない2番目の結果と、PMKが含まれている1番目の結果を比較すると、パケット397886は802.11 QoSデータとして復号化されます。

## 暗号化された802.11パケットの例

The screenshot shows a Wireshark capture of an 802.11 network. The packet list pane shows several packets, with packet 397886 highlighted in red. The packet details pane for packet 397886 shows the following structure:

- Frame 397886: 154 bytes on wire (1232 bits), 154 bytes captured (1232 bits)
- Radiotap Header v0, Length 48
- 802.11 radio information
- IEEE 802.11 QoS Data, Flags: .p....TC
- Data (68 bytes)

The packet bytes pane shows the raw data of the packet, with the QoS data field highlighted in blue.

注意：復号化時にWiresharkで問題が発生する可能性があり、その場合は、正しいPMKが指定されている（またはPSKが使用されている場合は、SSIDとPSKの両方が指定されている）場合でも、WiresharkはOTAキャプチャを復号化しません。この問題を回避するには、Wiresharkの電源をオフにしてから、上位レイヤ情報を取得して802.11パケットがQoSデータとして表示されなくなるまで数回使用するか、Wiresharkがインストールされている別のPC/Macを使用します。

ヒント: pmkXtractというC++コードが関連情報の最初の投稿に添付されています。コンパイラが正常に行われ、実行可能ファイルが取得されましたが、実行可能プログラムが正常に復号化を行っていないことが判明した理由があります。また、PMKを抽出しようとするPythonスクリプトは、最初の投稿のコメント領域に投稿されます。このコメント領域は、読者が興味を持っている場合に詳しく調べることができます。

## 関連情報

- [EAPの弱いリンクの調整 – pmkXtractを使用したWiFi PMKのRADIUSからの吸引](#)
- [RADIUS MS-MPPE-Recv-Keyのデコード方法](#)
- [テクニカル サポートとドキュメント – Cisco Systems](#)