Konfigurieren von Wireshark und FreeRADIUS zur Entschlüsselung des 802.11 WPA2-Enterprise/EAP/dot1x Wireless-Sniffers über die Funkverbindung

Inhalt

Einführung Voraussetzungen Anforderungen Verwendete Komponenten Hintergrundinformationen Vorgehensweise Schritt 1: Entschlüsseln von PMK(s) aus dem Access-Accept-Paket. Schritt 2: PMK(s) extrahieren. Schritt 3: Entschlüsseln Sie den OTA-Sniffer. Beispiel für ein entschlüsseltes 802.11-Paket Beispiel für ein verschlüsseltes 802.11-Paket Zugehörige Informationen

Einführung

In diesem Dokument wird die Entschlüsselung des verschlüsselten WLAN Protected Access 2 -Enterprise (WPA2-Enterprise)- oder 802.1x (dot1x)-verschlüsselten OTA-Sniffers (Wireless over the Air) mit allen Extensible Authentication Protocol (EAP)-Methoden beschrieben.

Es ist relativ einfach, PSK-basierte/WPA2-Personal 802.11 OTA-Erfassung zu entschlüsseln, solange die vollständigen EAP over LAN (EAPoL)-Handshakes erfasst werden. Pre-Shared Key (PSK) wird jedoch nicht immer aus Sicherheitsgründen empfohlen. Ein hartkodiertes Passwort zu knacken ist nur eine Frage der Zeit.

Daher entscheiden sich viele Unternehmen für dot1x mit Remote Authentication Dial-In User Service (RADIUS) als bessere Sicherheitslösung für ihr Wireless-Netzwerk.

Voraussetzungen

Anforderungen

Cisco empfiehlt, über Kenntnisse in folgenden Bereichen zu verfügen:

- FreeRADIUS mit installiertem Radsniff
- Wireshark/Omnipeek oder eine Software, die 802.11-Wireless-Datenverkehr entschlüsseln kann

- Berechtigung zum Abrufen des gemeinsam genutzten geheimen Codes zwischen Netzwerkzugriffsserver (NAS) und Authentifizierer
- Erfassung der Radius-Paketerfassung zwischen NAS und Authentifizierer von der ersten Zugriffsanforderung (von NAS zu Authentifizierer) bis zur letzten Zugriffsgenehmigung (vom Authentifizierer zu NAS) während der gesamten EAP-Sitzung
- Möglichkeit zur OTA-Erfassung (Over-the-Air) mit vierseitigen EAPoL-Handshakes

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf den folgenden Software- und Hardwareversionen:

- Radius-Server (FreeRADIUS oder ISE)
- Over-the-Air-Erfassungsgerät
- Apple MacOS/OS X- oder Linux-Gerät

Die Informationen in diesem Dokument wurden von den Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

Hintergrundinformationen

In diesem Beispiel werden zwei paarweise Master Keys (PMKs) von Radius-Paketen abgeleitet, die von ISE 2.3 erfasst wurden, da die Sitzungs-Timeout-Einstellung für diese SSID 1800 Sekunden beträgt und die hier angegebene Erfassung 34 Minuten (2040 Sekunden) lang ist.

Wie im Bild gezeigt, wird EAP-PEAP als Beispiel verwendet, kann jedoch auf jede dot1x-basierte Wireless-Authentifizierung angewendet werden.

]	wlan.	ddr==04:f1:28:6a:69:11 && (eapol or eap)		Expression	4
N	o.	Time	Source	Destination	Protocol Length Info	Ŀ
	43	5 2018-11-16 00:04:02.812197	Cisco_b4:3d:e4	HmdGloba_6a:69:11	EAP 109 Request, TLS EAP (EAP-TLS)	L
	43	7 2018-11-16 00:04:02.812927	HmdGloba_6a:69:11	Cisco_b4:3d:e4	EAP 73 Response, Legacy Nak (Response Only)	۰.
	43	9 2018-11-16 00:04:02.816752	Cisco_b4:3d:e4	HmdGloba_6a:69:11	EAP 109 Request, Protected EAP (EAP-PEAP)	I
	43	2 2018-11-16 00:04:02.818331	HmdGloba_6a:69:11	Cisco_b4:3d:e4	TLSVI.2 244 Client Hello	P
	434	9 2018-11-16 00:04:02.828460	Cisco_b4:3d:e4	HmdGloba_6a:69:11	TLSv1.2 1079 Server Hello, Certificate, Server Key Exchange, Server Hello	L
	43	2 2018-11-16 00:04:02.829281	HmdGloba_6a:69:11	Cisco_b4:3d:e4	EAP 73 Response, Protected EAP (EAP-PEAP)	L
	43	4 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	L
	43	6 2018-11-16 00:04:02.834110	HmdGloba_6a:69:11	Cisco_b4:3d:e4	EAP 73 Response, Protected EAP (EAP-PEAP)	L
	43	1 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	L
	43	3 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	L
	43	5 2018-11-16 00:04:02.851843	Cisco_b4:3d:e4	HmdGloba_6a:69:11	TLSv1.2 124 Change Cipher Spec, Encrypted Handshake Message	L
	43	7 2018-11-16 00:04:02.853063	HmdGloba_6a:69:11	Cisco_b4:3d:e4	EAP 73 Response, Protected EAP (EAP-PEAP)	Ŀ
1					8	

	A =	10	🎍 🛅	🗙 🖸	۹ .		1 i i i i i i i i i i i i i i i i i i i	1		୍ ୍	् 🎹	
--	------------	----	-----	-----	-----	--	---	---	--	-----	-----	--

	wlan.add	dr==04:f1:28:6a:69:11 && (eapol or eap))			🔀 📼 💌 Expression	÷
N	lo.	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, En	
	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 I	
	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)	¥
17	-						

Vorgehensweise

Schritt 1: Entschlüsseln von PMK(s) aus dem Access-Accept-Paket.

Führen Sie den **Radsniff** für die Radius-Erfassung zwischen NAS und Authenticator aus, um PMK zu extrahieren. Zwei Access-Accept-Pakete werden während der Erfassung extrahiert, weil der Sitzungs-Timeout-Timer für diese spezielle SSID auf 30 Minuten festgelegt ist und die Erfassung 34 Minuten lang ist. Die Authentifizierung erfolgt zweimal.

```
FRLU-M-51X5:pcaps frlu$ radsniff -I /Users/frlu/Downloads/radius_novlan_merged.pcapng -
{\rm 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
User-Name = "frlu_2"
State = 0x52656175746853657373696f6e3a306134323466326130303030303565373562656530393732
Class =
2f33303432
EAP-Message = 0x03c50004
Message-Authenticator = 0x38c67b9ba349842c9624889a45cabdfb
MS-MPPE-Send-Key = 0xa464cc15c0df8f09edc249c28711eb13a6db2d1a176f1196edcc707579fd6793
MS-MPPE-Recv-Key =
0xddb0b09a7d6980515825950b5929d02f236799f3e8a87f163c8ca41a066d8b3b<<<<<<<<<<<<>PMK
Authenticator-Field = 0x6cd33b4d4dde05c07d9923e17ad6c218
<snip>
2018-11-16 11:39:01.470000 (48) Access-Accept Id 183
/Users/frlu/Downloads/radius_novlan_merged.pcapng:10.66.79.42:32771 <- 10.66.79.36:1812 +0.000
+0.000
User-Name = "frlu_2"
State = 0x52656175746853657373696f6e3a30613432346632613030303030303565373562656530393732
Class =
2f33303434
EAP-Message = 0x03910004
Message-Authenticator = 0x81c572651679e15e54a900f3360c0aa9
MS-MPPE-Send-Key = 0xeae42cf7c6cd26371eee29856c51824fbb5bbb298874125928470114d009b5fb
MS-MPPE-Recv-Key =
0x7cce47eb82f48d8c0a91089ef7168a9b45f3d798448816a3793c5a4dfb1cfb0e<<<<<<<<<<>PMK
Authenticator-Field = 0xa523dd9ec2ce93d19fe4fc2e21537a5d
```

Hinweis: Entfernen Sie alle Virtual LAN (VLAN)-Tags der Radius-Paketerfassung. Andernfalls erkennt radsniff die pcap-Eingabedatei nicht. Zum Entfernen von VLAN-Tags kann beispielsweise <u>editcap</u> verwendet werden. **Tipp**: Im Allgemeinen kann die Laufzeit des Befehls **radsniff** für eine RADIUS pcap-Datei als Sekunden gezählt werden. Wenn der **Radsniff** jedoch in diesem im Protokoll angezeigten Zustand feststeckt, kaskadieren Sie diese Paketerfassung (A) mit einer weiteren längeren Paketerfassung (B) zwischen demselben NAS und Authentifizierer. Führen Sie dann den Befehl radsniff für das kaskadierte Paket (A+B) aus. Die einzige Anforderung für die Paketerfassung (B) besteht darin, dass Sie den Befehl radsniff darauf ausführen und das ausführliche Ergebnis anzeigen können.

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)

In diesem Beispiel wird die Protokollierung der Steuerungsebene (A), die über die <u>WLC-</u> <u>Paketprotokollierungsfunktion</u> erfasst wird, mit einer längeren Erfassung von TCPdump (B) der ISE kaskadiert. Die Protokollierung von WLC-Paketen dient als Beispiel, da sie in der Regel sehr klein ist.

WLC-Paketprotokollierung (A)

🚮 radius_novlan.pcap	Pcap Napture	22 KB Today at 11:56 am
ISE Tcpdump (B)		
adius_eap_decode_Cisco123.pcap	Yesterday at 12:04 pm	850 KB Pcap Napture
_		

Zusammengeführt (A+B)

🚋 radius_novlan_merged.pcapng

Führen Sie dann den **Radsniff** gegen das zusammengeführte pcap aus (A+B), und Sie können die ausführliche Ausgabe sehen.

Pcapn...Capture

927 KB Today at 12:28 pm

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>

Schritt 2: PMK(s) extrahieren.

Löschen Sie in jedem **MS-MPPE-Recv-Key** aus der ausführlichen Ausgabe das 0x-Feld, und die PMKs, die für die Wireless-Datenverkehrsdekodierung erforderlich sind, werden angezeigt.

MS-MPPE-Recv-Key = 0xddb0b09a7d6980515825950b5929d02f236799f3e8a87f163c8ca41a 066d8b3b

```
ddb0b09a7d6980515825950b5929d02f236799f3e8a87f163c8ca41a066d8b3b
```

```
MS-MPPE-Recv-Key = 0x7cce47eb82f48d8c0a91089ef7168a9b45f3d798448816a3793c5a4fb1cc
b0e
```

РМК: 7cce47eb82f48d8c0a91089ef7168a9b45f3d798448816a3793c5a4dfb1cfb0e

Schritt 3: Entschlüsseln Sie den OTA-Sniffer.

Navigieren Sie zu Wireshark > Preferences > Protocols > IEEE 802.11. Tippen Sie dann auf Entschlüsselung aktivieren und klicken Sie auf die Schaltfläche Bearbeiten neben Entschlüsselungsschlüssel, wie im Bild gezeigt.

HCrt A HDFS Reassemble fragmented 802.11 datagrams HDFSDATA Ignore vendor-specific HT elements HIP Ignore vendor-specific HT elements HSLIP Assume packets have FCS HNBAP Validate the FCS checksum if possible HP_ERM Ignore the Protection bit HSRP Yes - without IV HSRP Yes - with IV HTTP WPA Key MIC Length override IAPP WPA Key MIC Length override	A	Wireshark · Preferences	? ×
IAX2 IB ICAP ICAP ICEP ICP ICQ IEEE 802.11 IEEE 802.15,4 ↔ C > CK Cancel Help	HCrt ADFS HDFSDATA HIP HIQnet HISLIP HIT HNBAP HP_ERM HPFEEDS HSMS HSRP HTTP HTTP2 IAPP IAX2 IB ICAP ICEP ICMP ICP ICQ IEEE 802.11 IEEE 802.15.4 V	Image: Second	Help

Wählen Sie anschließend **wpa-psk** als Key-Typ aus, legen Sie die abgeleiteten PMKs in das **Key**-Feld ein, und klicken Sie dann auf **OK**. Nach Abschluss dieses Vorgangs sollte die OTA-Erfassung entschlüsselt werden, und Sie können Informationen auf höherer Ebene (3+) anzeigen.

Wireshark - Preferences	? X
WEP and WPA Decryption Keys ? X	
Key hype Key wpa-psk ddb0b09a7d6980515825950b5929d02/236799f3e8a87f163c8ca41a066d8b3b wpa-psk 7cce47eb82f48d8c0a91089ef7168a9b45f3d798448816a3793c5a4dfb1cfb0e	
+ - Pa A V B Citteers Hadnetzater Handlate Reaming littreshard (80211 Jans OK Cancel Help OK Cancel	Heb
	Wireshark - Preferences WEP and WPA Decryption Keys Yey: type Wpa-psk ddb0b09a7d6980515825950b5929d02/23679993e8a87/163c8ca41a066d8b3b wpa-psk rcce47eb82448d8c0a91099ef7168a9b4593d798448816a3793c5a4dtb1cfb0e + - Im Clikeers iddnesis trater ideoCata Reaming life edwrid (80211 ideos CK Cancel

Beispiel für ein entschlüsseltes 802.11-Paket

an.addr=	A . /. A		•				
	=04:f1:28:6a:69:11						Expression
	Time	Source		Destination	P	rotocol	Length Info
	397877 2018-11-16 00:17:08.095884	Cisco_b4:3d:e4	(00:a3:8e:b4:3d:e4) (T_	HmdGloba_6a:69:11 (04:f1:28	:6a:69:11) (RA) 8	02.11	45 Request-to-send, Flags=C
	397879 2018-11-16 00:17:08.097877	Cisco_b4:3d:e4	(00:a3:8e:b4:3d:e4) (T_	HmdGloba_6a:69:11 (04:f1:28	:6a:69:11) (RA) 8	02.11	45 Request-to-send, Flags=C
	397881 2018-11-16 00:17:08.098393	40.127.66.24		172.16.255.13	Т	CP	1438 [TCP Retransmission] 80 → 45658 [ACK] Seq=399990
	397882 2018-11-16 00:17:08.098444	104.17.57.239		172.16.255.13	T	CP	154 80 → 37553 [ACK] Seq=1 Ack=310 Win=65344 Len=0 T
	397883 2018-11-16 00:17:08.098495	HmdGloba_6a:69:	:11 (04:f1:28:6a:69:11)_	Cisco_b4:3d:e4 (00:a3:8e:b4	:3d:e4) (RA) 8	02.11	57 802.11 Block Ack, Flags=C
_	397884 2018-11-16 00:17:08.098999	104.17.57.239		172.16.255.13	T	CP	162 80 → 37555 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=
	397886 2018-11-16 00:17:08.099099	172.16.255.13		40.127.66.24	T	CP	154 45658 → 80 [ACK] Seq=128 Ack=4001196 Win=788480
_	39/88/ 2018-11-16 00:1/:08.099181	Cisco_b4:3d:e4	(00:a3:8e:b4:3d:e4) (1_	HmdGloba_6a:69:11 (04:11:28	:6a:69:11) (RA) 8	02.11	57 802.11 Block Ack, Flags=
	397888 2018-11-16 00:17:08.099606	172.16.255.13		104.17.57.239	T	CP	154 37555 → 80 [ACK] Seq=1 Ack=1 Win=87808 Len=0 TSv
	397889 2018-11-16 00:17:08.099655	Cisco_b4:3d:e4	(00:a3:8e:b4:3d:e4) (T_	HmdGloba_6a:69:11 (04:f1:28	:6a:69:11) (RA) 8	02.11	57 802.11 Block Ack, Flags=C
	397890 2018-11-16 00:17:08.101762	172.16.255.13		104.17.57.239	H	ITTP	479 GET /s100264/images/logoq.png?t=636366 HTTP/1.1
	397891 2018-11-16 00:17:08.101812	Cisco_b4:3d:e4	(00:a3:8e:b4:3d:e4) (T_	HmdGloba_6a:69:11 (04:†1:28	:6a:69:11) (RA) 8	02.11	57 802.11 Block Ack, Flags=C
ie 39	7886: 154 bytes on wire (1232 bits),	154 bytes captu	red (1232 bits)				
iotap	Header v0, Length 48						
.11 r	adio information						
E 802	.11 QoS Data, Flags: .pTC						
ical-	Link Control						
ernet	Protocol Version 4, Src: 172.16.25	.13, Dst: 40.127	.66.24				
	sion Control Protocol, Src Port: 450	58, Dst Port: 80	, Seq: 128, Ack: 4001190	5, Len: 0			
121172			· · · · · · · · · · · · · · · · · · ·				
120122							
120122							
15#115							
124172							
124112							
120122							
15#13							
00 0	0 30 00 65 08 1c 00 6d f9 30 31 00	80 00 00 ···0·k	··· m·01····				
00 0 14 0	0 30 00 65 08 1c 00 6d f9 30 31 00 0 9e 09 80 04 d9 a4 00 00 00 80	80 00 00 ···0·k· 84 01 00 ·····					
00 0 14 0 9e 0	0 30 00 65 08 12 00 6d f9 30 31 00 9 96 09 80 64 d9 a4 00 00 00 08 80 9 95 22 1f 00 06 00 65 00 08 04	00 00 00 ···0·k 04 01 00 ····- 00 00 00 ···	m=01····				
00 0 14 0 9e 0 88 4	0 30 00 55 06 12 00 6d f9 30 31 00 9 9e 09 80 04 d9 a4 00 00 00 80 9 05 22 1f 00 05 00 65 00 00 00 04 1 30 00 00 a3 Be b4 30 44 04 f1 26	00 00 00 ··0·k 04 01 00 06 00 00 ···· 5a 69 11 ·A0·	e				
00 0 14 0 9e 0 88 4 00 0	0 30 00 6b 08 1c 00 6d f9 30 31 00 0 9e 09 80 04 d9 a4 00 00 00 08 9 0b 22 1f 00 66 00 65 00 00 00 04 1 30 00 00 a3 8e b4 3d e4 04 f1 28 c 29 28 89 dd 50 06 00 08 84 00	00 00 00 ···0·k 04 01 00 00 00 00 ···· 20 01 00 ··)(··	e • (ji ₽				
00 0 14 0 9e 0 88 4 00 0	0 30 00 6b 08 1c 00 6d f9 30 31 00 0 9e 09 80 04 d9 a4 00 00 00 00 80 0 9b 22 1f 00 65 00 55 00 00 00 04 1 30 00 00 a1 3e b4 3d a4 04 11 28 c 29 28 89 dd 50 06 00 00 c 88 4 00 c 39 f 4c 22 f 90 d1 14 52 a5 6b 2c	00 00 00 00 00 k 00 00 00 00 00 00 00 00 00 00 00 00 00					
00 0 14 0 9e 0 88 4 00 0 d8 5	0 30 00 6b 08 1c 00 6d f9 30 31 00 0 9e 09 80 04 d9 a4 00 00 00 08 9 9b 22 1f 00 66 00 65 00 00 00 41 13 00 00 03 16 b4 3d 64 04 f1 28 c 29 28 09 dd 50 06 00 00 c8 84 00 a ff 4c 22 f9 04 11 42 c3 5b c 4 55 56 a 12 92 da fc 99 1f c2 c8 7 7 26 57 40 c4 c4 c7 28	00 00 00					

Wenn Sie das zweite Ergebnis vergleichen, bei dem der PMK nicht enthalten ist, mit dem ersten Ergebnis, bei dem der PMK enthalten ist, wird Paket 397886 als 802.11-QoS-Daten entschlüsselt.

Beispiel für ein verschlüsseltes 802.11-Paket

	*U4:T1:28:68:69:11				
	Time	Source	Destination	Protocol	Length Info
	397881 2018-11-16 00:17:08.098393	Vmware_28:89:dd	HmdGloba_6a:69:11	802.11	1438 QoS Data, SN=1434, FN=0, Flags=.pR.F.C
	397882 2018-11-16 00:17:08.098444	Vmware_28:89:dd	HmdGloba_6a:69:11	802.11	154 QoS Data, SN=1435, FN=0, Flags=.pF.C
	397883 2018-11-16 00:17:08.098495	HmdGloba_6a:69:11 (04:f1:28:6a:69:11)_	Cisco_b4:3d:e4 (00:a3:8e:b4:3d:e4) (RA)	802.11	57 802.11 Block Ack, Flags=C
	397884 2018-11-16 00:17:08.098999	Vmware_28:89:dd	HmdGloba_6a:69:11	802.11	162 QoS Data, SN-1436, FN-0, FlagspF.C
	397886 2018-11-16 00:17:08.099099	HmdGloba_6a:69:11	Vmware_28:89:dd	802.11	154 QoS Data, SN=101, FN=0, Flags=.pTC
	397887 2018-11-16 00:17:08.099181	Cisco_b4:3d:e4 (00:a3:8e:b4:3d:e4) (T.	HmdGloba_6a:69:11 (04:f1:28:6a:69:11) (RA)	802.11	57 802.11 Block Ack, Flags=
	397888 2018-11-16 00:17:08.099606	HmdGloba_6a:69:11	Vmware_28:89:dd	802.11	154 QoS Data, SN=102, FN=0, Flags=.pTC
	397889 2018-11-16 00:17:08.099655	Cisco_b4:3d:e4 (00:a3:8e:b4:3d:e4) (T_	HmdGloba_6a:69:11 (04:f1:28:6a:69:11) (RA)	802.11	57 802.11 Block Ack, Flags=C
	397890 2018-11-16 00:17:08.101762	HmdGloba_6a:69:11	Vmware_28:89:dd	802.11	479 QoS Data, SN=103, FN=0, Flags=.pTC
	397891 2018-11-16 00:17:08.101812	Cisco_b4:3d:e4 (00:a3:8e:b4:3d:e4) (T_	HmdGloba_6a:69:11 (04:f1:28:6a:69:11) (RA)	802.11	57 802.11 Block Ack, Flags=C
	397892 2018-11-16 00:17:08.105958	Vmware_28:89:dd	HmdGloba_6a:69:11	802.11	595 QoS Data, SN=1437, FN=0, Flags=.pF.C
	397894 2018-11-16 00:17:08.106056	Vmware 28:89:dd	HmdGloba_6a:69:11	802.11	154 QoS Data, SN=1438, FN=0, Flags=.pF.C
1000 00 00 010 14 00 028 9e 09	130 00 65 06 1c 00 6d f9 30 31 00 9e 09 80 04 d9 a4 00 00 00 80 00 05 22 1f 00 66 00 65 80 06 00 01	00 00 00 · · 0·k· · m·01···· 04 01 00 · · · · · · e · · · · ·			
000 00 00 110 14 00 120 9 09 130 88 41 140 00 00 150 88 41	130 00 6b 08 1c 00 6d f9 30 31 00 19e 09 30 04 d9 a4 00 00 00 00 80 0b 22 1f 00 66 00 65 00 00 00 43 30 00 00 a3 1e 14 34 e4 04 f1 28 29 28 69 dd 50 06 00 c8 64 00 r f4, 2 7 6 a4 11 45 a5 8b, 2	00 00 00 · 0 · k · · m · 01 · · · · · · · · · · · · · · · · ·			
360 00 00 00 915 14 00 32.2 92 99 93.0 88 41 94.0 00 <	130 00 65 08 1c 00 6d f9 30 31 00 9e 09 68 04 d9 a4 00 08 00 08 00 06 22 1f 00 06 00 65 00 00 08 0 30 00 10 a3 8e b4 30 e4 04 f1 28 99 28 89 d5 86 00 oc 68 24 00 af f4 c2 2f 90 d1 14 52 a5 8b 2e a5 55 0a 12 92 da f ca 91 ft c2 c6	00 00 000 k m 01 04 01 00			
000 00 00 010 14 00 020 92 99 030 88 41 040 00 00 050 08 90 060 48 54 050 08 54	30 00 65 08 1.c 00 6d f9 30 31 00 9e 09 80 04 49 40 00<	00 00 00 00 00 00 00 00 00 00 00 00 00			

Vorsicht: Sie können bei der Entschlüsselung auf ein Problem mit Wireshark stoßen. Selbst wenn der richtige PMK bereitgestellt wird (oder wenn PSK verwendet wird, werden sowohl SSID als auch PSK bereitgestellt), entschlüsselt Wireshark die OTA-Erfassung nicht. Die Problemumgehung besteht darin, Wireshark ein- und mehrmals auszuschalten, bis höhere Layer-Informationen abgerufen werden können und 802.11-Pakete nicht mehr als QoS-Daten angezeigt werden, oder einen anderen PC/Mac zu verwenden, auf dem Wireshark installiert ist.

Tipp: Ein C++-Code namens pmkXtract ist im ersten Beitrag in Related Information angehängt. Die zu kompilierenden Versuche wurden erfolgreich durchgeführt, und es wird eine ausführbare Datei abgerufen, aber das ausführbare Programm scheint die Entschlüsselung aus einigen unbekannten Gründen nicht ordnungsgemäß durchzuführen. Zusätzlich wird ein Python-Skript, das versucht, PMK zu extrahieren, im Kommentarbereich des ersten Beitrags veröffentlicht, der bei Interesse der Leser weiter erforscht werden kann.

Zugehörige Informationen

- Optimierung der schwachen Verbindung von EAP Entziehen von Wi-Fi-PMKs aus RADIUS mit pmkXtract
- Dekodieren des Radius MS-MPPE-Recv-Key
- <u>Technischer Support und Dokumentation Cisco Systems</u>