

Scuola Politecnica e
delle Scienze di Base



Università degli Studi di Napoli Federico II

Facoltà di Ingegneria delle Telecomunicazioni

KRACK Attack

Network Security - Bianca Ruggiero M59/167



WPA/WPA2

Protocollo di sicurezza introdotto nel 2003 nell' *amendment* di 802.11i per ovviare alla vulnerabilità del protocollo WEP.

L'*amendment* ha definito:

- **il 4-way-handshake**
- **Wi-Fi Protected Access-Temporal Key Integrity Protocol (WPA-TKIP)**
- **Advanced Encryption Standard-CCMP (AES-CCMP))**

Il protocollo WPA/WPA2 è considerato formalmente sicuro.



4-Way-Handshake

Algoritmo necessario per generare le chiavi temporanee per la crittografia, comincia dopo la fase di autenticazione ed associazione del client all'AP.

Vengono generate due tipi di chiave:

- **Pairwise Transient Key (PTK)**

$$f(PMK, A \text{Nonce}, S \text{Nonce}, A \text{MAC}, S \text{MAC})$$

PMK = Pairwise Master Key, segreto pre-condiviso

viene poi suddivisa in:

- Key Confirmation Key (KVK)
- Key Encryption Key (KEK)
- Temporal Key (TK)

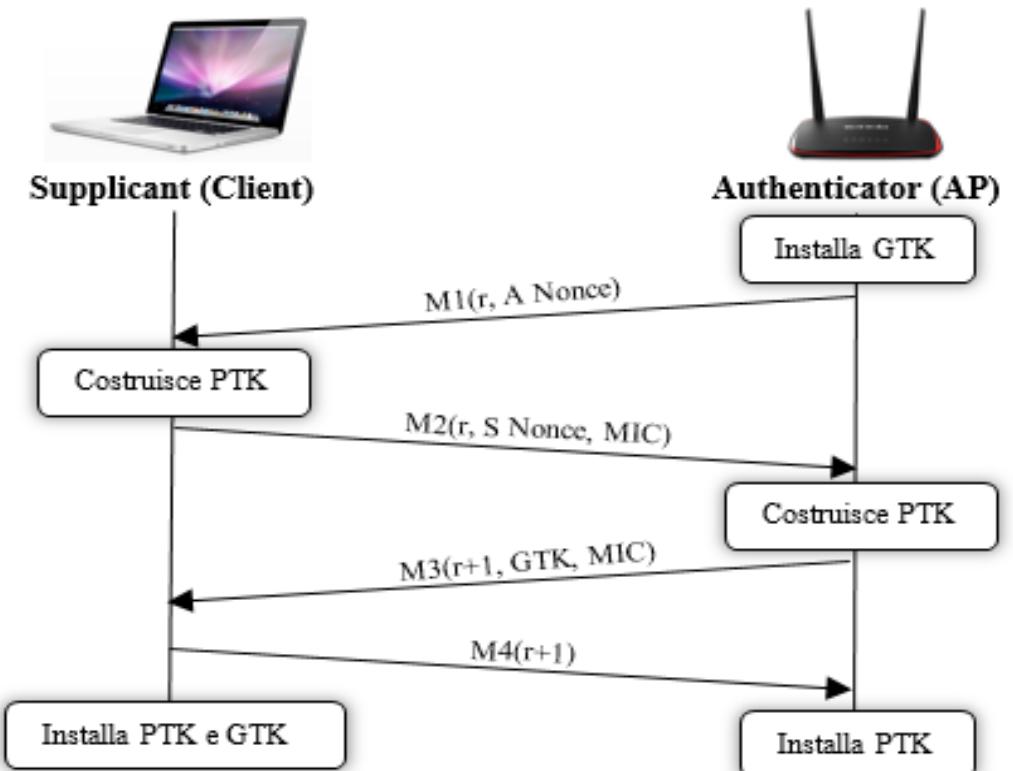
- **Group Temporal Key (GTK)**

$$f(GMK, A \text{Address}, G \text{Nonce})$$

4-Way-Handshake

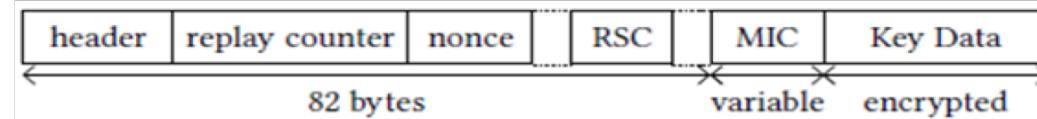
La PTK che viene generata, può essere aggiornata sempre attraverso un nuovo 4-way-handshake, dove però tutti i messaggi sono criptati utilizzando il PTK corrente.

Inoltre è possibile che si possano perdere le frame e dunque il client si troverà a gestire la ritrasmissione dei messaggi in particolare del primo e del terzo.





Frame EAPOL



La parte di dimensione fissa (82 bytes) è composta:

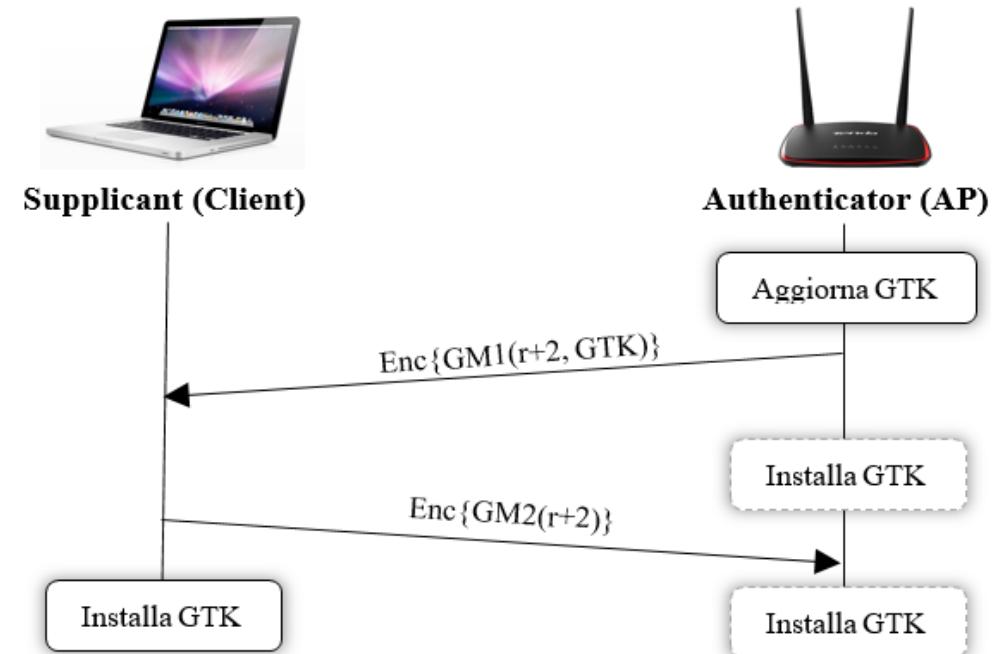
- **header**, identifica quale messaggio viene inviato
- **replay counter**, contatore incrementale utilizzato per rilevare eventuali frame ripetute
- **nonce**, nonce casuale che genera singolarmente il supplicant e l'authenticator.
- **receiver sequence counter (RSC)**, contiene il numero di pacchetto iniziale della GTK, se prevista.

La parte a dimensione variabile si compone dei campi:

- **key data**, contiene la GTK ed eventuali altri parametri, campo protetto dalla KEK.
- **Message integrity check (MIC)**, utilizzato per la verifica di autenticità della frame che viene protetta dalla KCK.

Group Handshake

La GTK viene aggiornata periodicamente dall'AP, che la distribuisce ai client connessi ad esso. Da notare che la GTK, contenuta nel campo key data della frame EAPOL, è protetta dalla KEK, legata alla PTK corrente.





Protocolli di integrità e riservatezza dei dati

➤ Temporal Key Integrity Protocol (TKIP)

- La TK viene suddivisa in una chiave di 128 bit e due chiavi MIC rispettivamente usate per i due versi di comunicazione.
- Viene usato RC4 per la crittografia che usa la chiave di 128 bit, l'indirizzo MAC del mittente e un nonce di 48 bit.
- Il nonce viene incrementato per ogni frame trasmessa e viene inizializzato ad 1 dopo ogni installazione della TK, viene usato anche come replay counter.
- Protocollo vulnerabile



Protocolli di integrità e riservatezza dei dati

➤ AES-CCMP

- Protocollo basato su cifrario a blocchi
- L'algoritmo crittografico che opera in modalità Counter Mode (CTR), viene accoppiato a un codice di autenticazione MAC (CBC-MAC) .
- Utilizza un vettore di inizializzazione (IV) una concatenazione: indirizzo MAC mittente, nonce di 48 bit e flag aggiuntivi legati alla frame trasmessa
- Nonce incrementato per ogni frame trasmessa ed inizializzato 0, viene usato anche come replay counter.
- Protocollo più sicuro e più diffuso



Key Reinstallation Attack – 4-Way-Handshake

Requisiti:

- L'attaccante deve essere fisicamente presente e in posizione di Man-in-the-Middle (MitM), sul canale, tra il Client e l'AP.
- La vittima deve accettare la ritrasmissione della frame 3, nel 4-way-handshake
 - In chiaro
 - Protette dal protocollo di integrità e riservatezza dei dati, una volta installata la PTK.
- L'AP deve accettare messaggi di risposta con replay counter precedenti, non ancora ricevuti.
- Il client, nel caso riceve un messaggio 3 ritrasmesso crittografato con una PTK, non deve controllare se è stata effettivamente aggiornata.



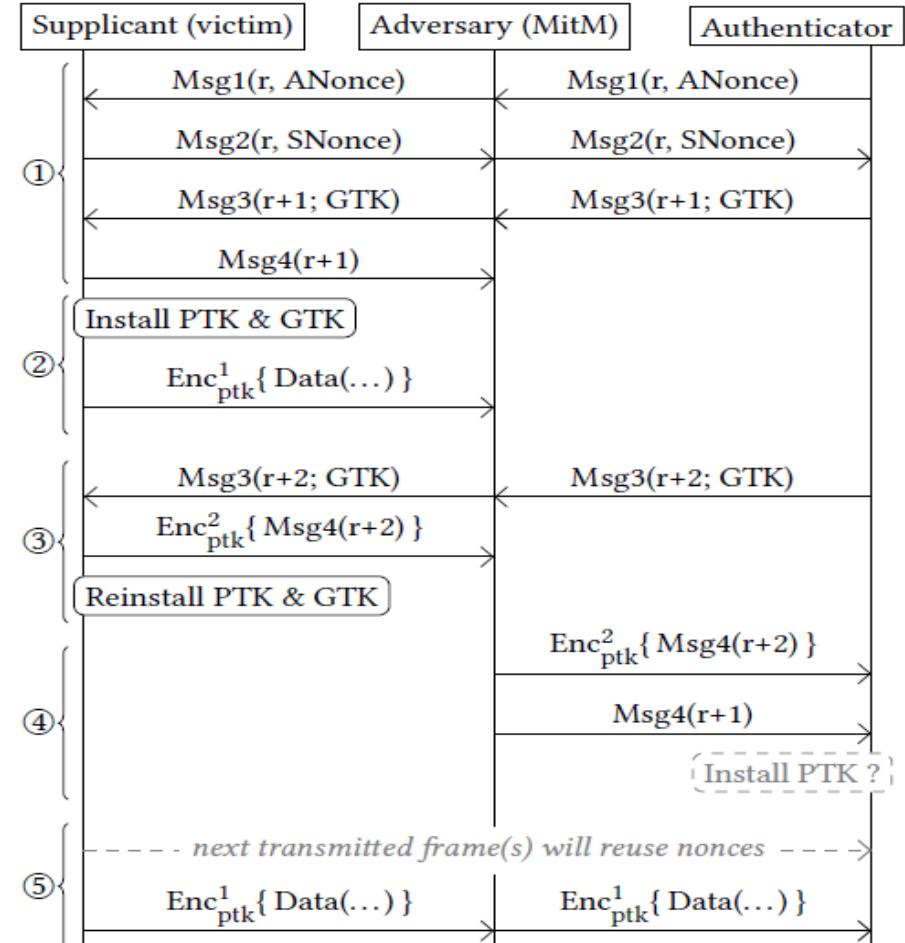
KRACK Attack

Key Reinstallation Attack – 4-Way-Handshake

La vittima accetta ritrasmissioni in chiaro del messaggio 3, dopo aver installato la PTK

Nota:

- L'attacco si può ripetere più volte, in quanto basta de-autenticare la vittima in modo da forzare un nuovo 4-way-handshake.
- La reinstallazione della PTK può verificarsi anche senza attaccante, a causa del rumore sul canale, per cui si può pensare anche ad un attacco jam.





Key Reinstallation Attack – 4-Way-Handshake

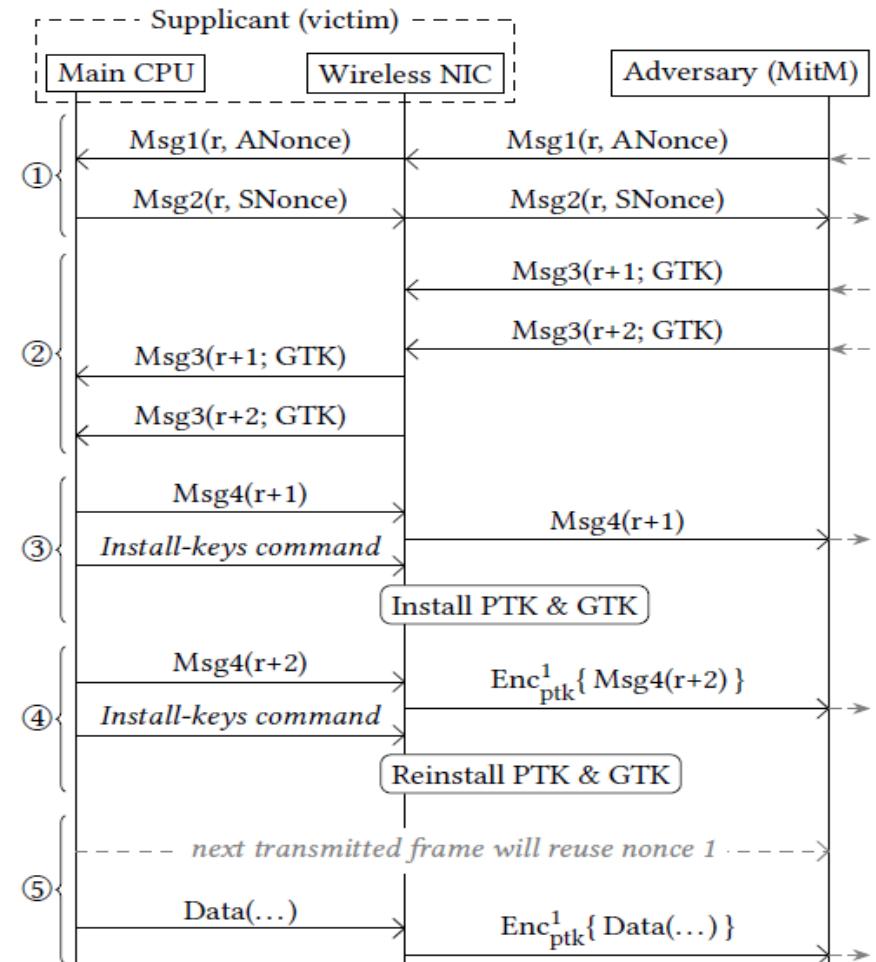
La vittima accetta ritrasmissioni in chiaro del messaggio 3, dopo aver installato la PTK, solo se ricevute immediatamente una dopo l'altra.

(Supplicant Android e Linux)

- Wireless NIC: implementa il protocollo di data-confidentiality.
- Main CPU: implementa il 4-way-handshake

Gli scambi tra attaccante e Authenticator sono gli stessi del caso precedente.

KRACK Attack





Key Reinstallation Attack – 4-Way-Handshake

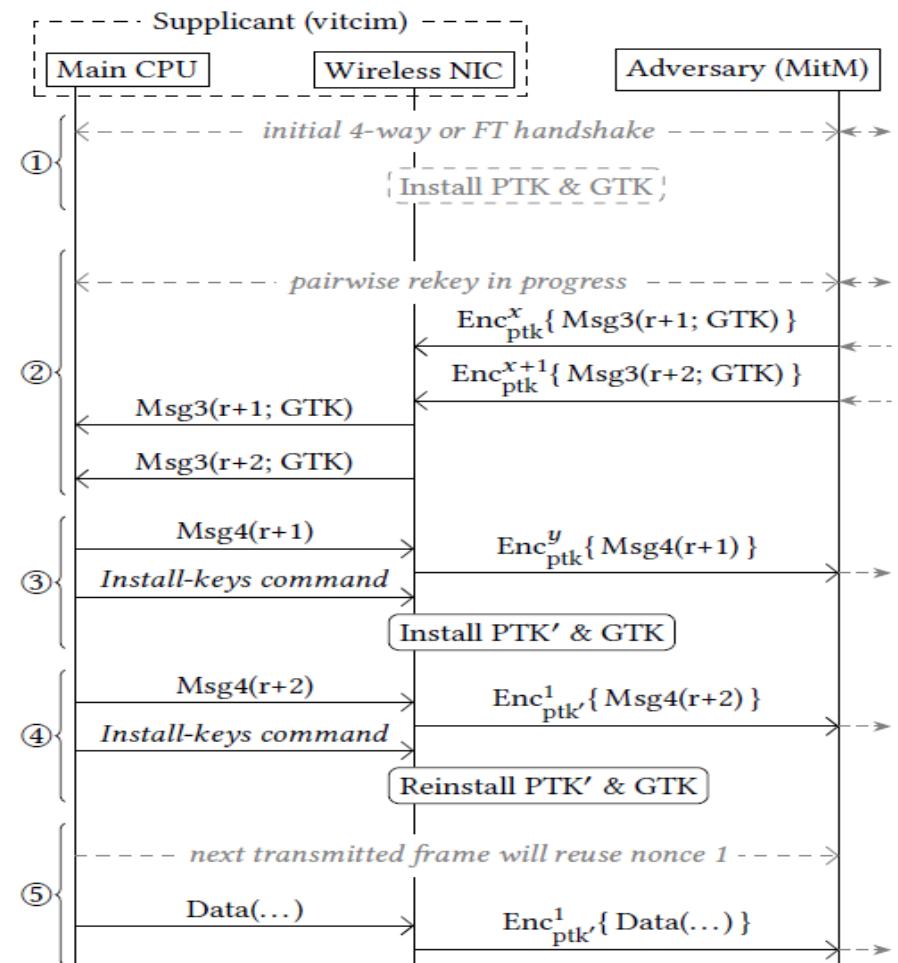
**La vittima accetta ritrasmissioni crittografate del messaggio 3, dopo aver installato la PTK.
(OpenBSD, OS X, macOS)**

- Wireless NIC: implementa il protocollo di data-confidentiality.
- Main CPU: implementa il 4-way-handshake

Gli scambi tra attaccante e Authenticator sono gli stessi del caso precedente.

L'attacco si fa in fase di rekey, che si può facilmente forzare.

KRACK Attack





Key Reinstallation Attack – Group Handshake

Requisiti:

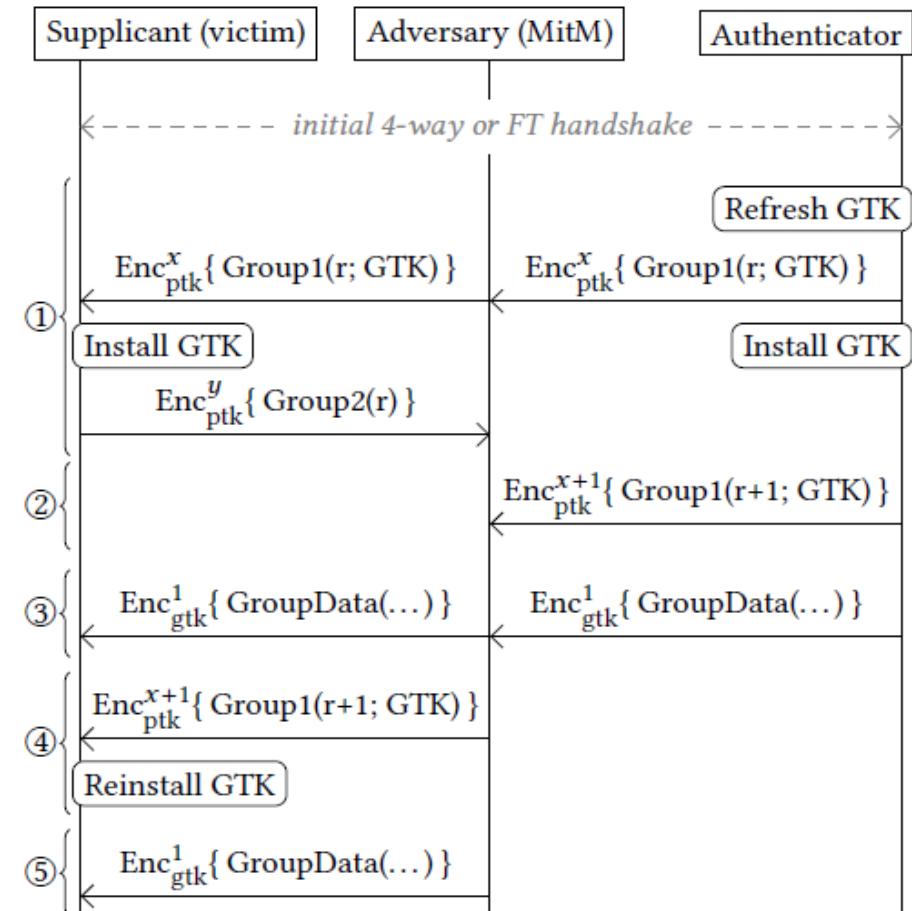
- L'attaccante deve essere fisicamente presente e in posizione di Man-in-the-Middle (MitM), sul canale, tra il Client e l'AP.
- La vittima deve re-inizializzare il replay counter durante l'installazione della GTK.
- L'attaccante deve riuscire a catturare il messaggio di gruppo 1, idoneo per il client e che contenga la chiave di gruppo già installata dall'AP.
- L'AP può installare la GTK:
 - subito dopo inviato il messaggio di gruppo 1
 - dopo aver ricevuto, la risposta dal client (complica un po' l'attacco)

L'attacco si può facilmente forzare



Key Reinstallation Attack – Group Handshake

L'AP installa la GTK subito dopo aver inviato il messaggio di gruppo 1.





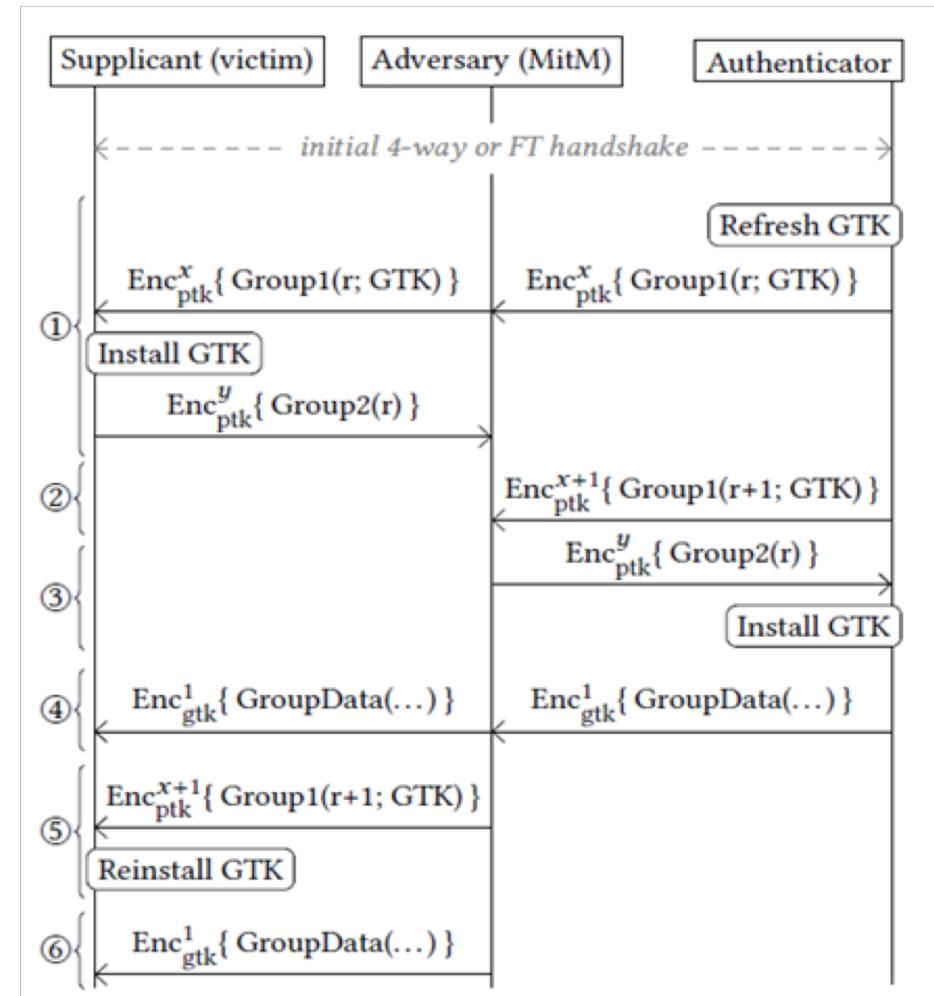
KRACK Attack

Key Reinstallation Attack – Group Handshake

L'AP installa la GTK in modalità ritardata, ovvero dopo aver ricevuto il messaggio di gruppo 2 dal client.

È necessario che l'AP accetti messaggi con replay counter antecedenti a quello corrente.

OpenBSD non è vulnerabile perché installa la GTK in modalità ritardata e accetta solo messaggi con replay counter corrente.



Key Reinstallation Attack – Fast Basic Service Set Transition Handshake

Definito in 802.11r, per ridurre il tempo di roaming del Client tra un AP e un altro sulla stessa rete protetta.

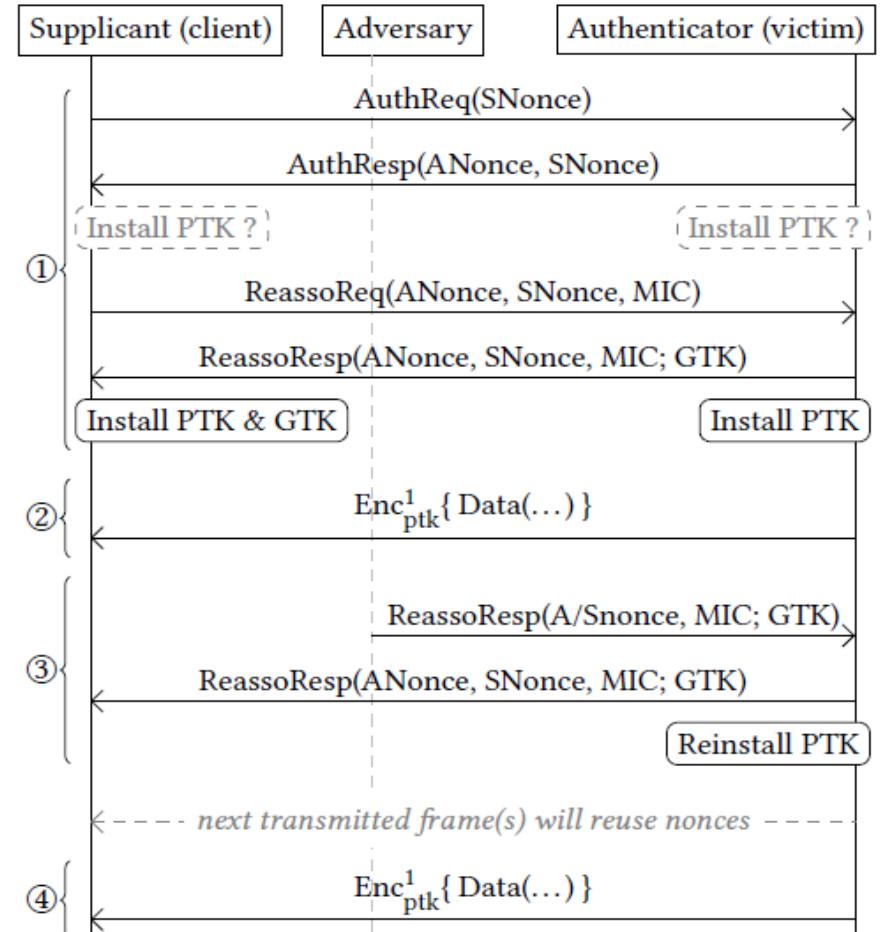
Nota:

La PTK dovrebbe essere installata dopo i messaggi di autenticazione, invece viene installata alla fine dell'handshake.

Attacco devastante: il messaggio di ri-associazione può essere ritrasmesso perché non c'è il replay counter.

Non è necessaria la posizione MitM

KRACK Attack





Key Reinstallation Attack – Impatto

- Il riutilizzo del nonce, implica il riutilizzo del keystream, che porta a decifrare le frame.
- La ripetizione del replay counter, permette attacchi di tipo replay.
- Se si usa TKIP è possibile recuperare il MIC, e di conseguenza forgiare frame nella direzione che parte dalla vittima :
 - il client, nel caso del 4-way-handshake
 - l'AP nel caso del FT handshake.
- Se si usa GCMP, (usato per Wireless Gigabit) si riesce a recuperare il MIC per entrambe le direzioni.
- In wpa-supplicant 2,4 e 2,5, l'attacco produce la reinstallazione di una chiave di crittografia con tutti zero.



Key Reinstallation Attack – Contromisure

- Il protocollo di data-confidentiality dovrebbe verificare se è incorso la reinstallazione di una PTK già in uso, e nel caso non reimpostare il nonce e/o il replay counter associato.
- Assicurarsi che la chiave sia installata una sola volta nella Wireless NIC, durante l'handshake.
- Essere più rigorosi nella descrizione e nell'implementazione degli standard.
- Installare sempre gli ultimi aggiornamenti proposti dai rispettivi produttori.



Test dispositivi

I dispositivi, gli applicativi e i sistemi operativi vulnerabili sono vari, anche se ognuno implementa lo standard in modo diverso.

Si sottopongono al test 3 dispositivi:

- Samsung Galaxy S4, GT-I9505-Versione Android 5.0.1
- OUKITEL, U11 Plus – Versione Android 7.0
- Iphone 4S, MD239B/A – Versione iOS 9.3.5 (13G36)

Il test che viene effettuato è il `./krack-test-client.py`, che testa la reinstallazione delle chiavi nel 4-way-handshake sia della PTK sia della GTK.

Implementation	Re. Msg3	Pt. EAPOL	Quick Pt.	Quick Ct.	4-way	Group
OS X 10.9.5	✓	✗	✗	✓	✓	✓
macOS Sierra 10.12	✓	✗	✗	✓	✓	✓
iOS 10.3.1 ^c	✗	N/A	N/A	N/A	✗	✓
wpa_supplicant v2.3	✓	✓	✓	✓	✓	✓
wpa_supplicant v2.4-5	✓	✓	✓	✓ ^a	✓ ^a	✓
wpa_supplicant v2.6	✓	✓	✓	✓ ^b	✓ ^b	✓
Android 6.0.1	✓	✗	✓	✓ ^a	✓ ^a	✓
OpenBSD 6.1 (rum)	✓	✗	✗	✗	✗	✓
OpenBSD 6.1 (iwn)	✓	✗	✗	✓	✓	✓
Windows 7 ^c	✗	N/A	N/A	N/A	✗	✓
Windows 10 ^c	✗	N/A	N/A	N/A	✗	✓
MediaTek	✓	✓	✓	✓	✓	✓

^a Due to a bug, an all-zero TK will be installed, see Section 6.3.

^b Only the group key is reinstalled in the 4-way handshake.

^c Certain tests are irrelevant (not applicable) because the implementation does not accept retransmissions of message 3.



Test dispositivi – Configurazione Kali Linux

```
root@Ruggiero:~# apt-get update

root@Ruggiero:~# apt-get install libnl-3-dev libnl-genl-3-dev pkg-config libssl-dev net-tools git sysfsutils python-scapy python-pycryptodome

root@Ruggiero:~# git clone https://github.com/vanhoefm/krackattacks-scripts.git

root@Ruggiero:~/krackattacks-scripts/krackattack# ./disable-hwcrypto.sh
Done. Reboot your computer.
root@Ruggiero:~/krackattacks-scripts/krackattack# reboot

root@Ruggiero:~# rfkill unblock wifi

root@Ruggiero:~/krackattacks-scripts/hostapd# cp defconfig .config
root@Ruggiero:~/krackattacks-scripts/hostapd# make -j 2
```



Test dispositivi – Lancio dello script

```
root@Ruggiero:~/krackattacks-scripts/krackattack# ./krack-test-client.py
[12:16:22] Note: disable Wi-Fi in network manager & disable hardware encryption.
Both may interfere with this script.
[12:16:22] Starting hostapd ...
Configuration file: /root/krackattacks-scripts/krackattack/hostapd.conf
Using interface wlan0 with hwaddr 48:5d:60:c4:db:69 and ssid "testnetwork-maggio
2018"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
[12:16:23] Ready. Connect to this Access Point to start the tests. Make sure the
client requests an IP using DHCP!
[12:16:24] Reset PN for GTK
[12:16:26] Reset PN for GTK
[12:16:28] Reset PN for GTK
[12:16:31] Reset PN for GTK
[]
```



Test dispositivi – Configurazione per il monitoraggio

```
root@Ruggiero:~# airmon-ng
PHY      Interface     Driver      Chipset
phy0      wlan0         ath9k
          Adapter (PCI-Express) (rev 01)
phy1      wlan1         rtl8187
          realtek RTL8187B]

root@Ruggiero:~# airmon-ng check kill
Killing these processes:
PID Name
525 wpa_supplicant

root@Ruggiero:~# airmon-ng start wlan1
root@Ruggiero:~# iwconfig wlan1mon channel 6
root@Ruggiero:~# airodump-ng wlan1mon --channel 6

CH 6 ][ Elapsed: 6 s ][ 2018-05-30 19:06
BSSID          PWR RXQ  Beacons   #Data, #/s CH MB ENC CIPHER AUTH ESSID
48:5D:60:C4:DB:69 -16 100      69       0    0   6 54e WPA2 CCMP  PSK testnetwork-maggio2018
BSSID          STATION          PWR  Rate   Lost   Frames Probe
(not associated) 3C:CF:58:AE:58:36 -65    0 - 1     1      4
```



Test dispositivi – I° Dispositivo

cattura s4.pcapng

File Modifica Visualizza Val Cattura Analizza Statistiche Telefonia Wireless Strumenti Aiuto

Aplica un filtro di visualizzazione ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
118	5.638671389	MurataMa_25:d5:f2	Azureway_c4:db:69	802.11	54	Null function (No data), SN=2193, FN=0, Flags=.....TC
119	5.638862878	MurataMa_25:d5:f2 (f...	802.11	48	Acknowledgement, Flags=.....C	
120	5.641114974	MurataMa_25:d5:f2	Azureway_c4:db:69	802.11	179 QoS Data, SN=3, FN=0, Flags=p....TC	
121	5.641478624	MurataMa_25:d5:f2 (f...	802.11	48	Acknowledgement, Flags=.....C	
122	5.641978968	MurataMa_25:d5:f2	Azureway_c4:db:69	802.11	54 Null function (No data), SN=2194, FN=0, Flags=.....TC	
123	5.642353264	MurataMa_25:d5:f2 (f...	802.11	48	Acknowledgement, Flags=.....C	
124	5.734456428	Azureway_c4:db:69	Broadcast	802.11	188 Beacon frame, SN=3056, FN=0, Flags=.....C, BI=100, SSID=testnetwor	
125	5.836757569	Azureway_c4:db:69	Broadcast	802.11	188 Beacon frame, SN=3057, FN=0, Flags=.....C, BI=100, SSID=testnetwor	
126	5.921674128	MurataMa_25:d5:f2	Broadcast	802.11	422 QoS Data, SN=4, FN=0, Flags=p....TC	
127	5.921674128	MurataMa_25:d5:f2	Broadcast	802.11	100 Data, SN=100, FN=0, Flags=.....C	

Frame Control Field: 0x8841
.000 0001 0011 1010 = Duration: 314 microseconds
Receiver address: Azureway_c4:db:69 (48:5d:60:c4:db:69)
Destination address: Azureway_c4:db:69 (48:5d:60:c4:db:69)
Transmitter address: MurataMa_25:d5:f2 (f0:27:65:25:d5:f2)
Source address: MurataMa_25:d5:f2 (f0:27:65:25:d5:f2)
BSS Id: Azureway_c4:db:69 (48:5d:60:c4:db:69)
STA address: MurataMa_25:d5:f2 (f0:27:65:25:d5:f2)
.... 0000 = Fragment number: 0
0000 0000 0011 = Sequence number: 3
Frame check sequence: 0x47abc9b2 [correct]
[FCS Status: Good]
Qos Control: 0x0000
CCMP parameters
CCMP Ext. Initialization Vector: 0x000000000001
Key Index: 0
Data (115 bytes)

```
0000 00 00 1a 00 2f 40 00 00 2e b9 5c 10 00 00 00 00 .....H.. .\.
0010 10 02 85 09 a0 00 db 01 00 00 08 41 3a 01 48 50 ..... .A; H]
0020 68 c4 db 69 f0 27 65 25 d5 f2 48 5d 00 c4 db 69 ..i.'% ..H] ..i
```

Frame (frame), 179 byte

Pacchetti: 7291 · visualizzati: 7291 (100.0%) · Tempo di caricamento: 0:0.187 · Profilo: Default

```
[14:59:25] Reset PN for GTK
[14:59:25] f0:27:65:25:d5:f2: sending a new 4-way message 3 where the GTK has a zero RSC
[14:59:25] f0:27:65:25:d5:f2: DHCP reply 192.168.100.2 to f0:27:65:25:d5:f2
[14:59:27] Reset PN for GTK
[14:59:27] f0:27:65:25:d5:f2: sending a new 4-way message 3 where the GTK has a zero RSC
[14:59:27] f0:27:65:25:d5:f2: client has IP address -> now sending replayed broadcast ARP packets
[14:59:27] f0:27:65:25:d5:f2: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 0 ARPs this interval)
[14:59:27] f0:27:65:25:d5:f2: IV reuse detected (IV=1, seq=19). Client reinstalls the pairwise key in the 4-way handshake (this is bad)
[14:59:29] Reset PN for GTK
[14:59:29] f0:27:65:25:d5:f2: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 1 ARPs this interval)
[14:59:31] Reset PN for GTK
[14:59:31] f0:27:65:25:d5:f2: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 2 ARPs this interval)
[14:59:31] f0:27:65:25:d5:f2: sending broadcast ARP to 192.168.100.1 (sent 3 ARPs this interval)
[15:00:20] f0:27:65:25:d5:f2: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 2 ARPs this interval)
[15:00:22] Reset PN for GTK
[15:00:22] f0:27:65:25:d5:f2: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 3 ARPs this interval)
[15:00:24] Reset PN for GTK
[15:00:24] f0:27:65:25:d5:f2: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 4 ARPs this interval)
[15:00:26] Reset PN for GTK
[15:00:26] f0:27:65:25:d5:f2: Client DOESN'T reinstall the group key in the 4-way handshake (this is good)
[15:00:26] f0:27:65:25:d5:f2: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 1 ARPs this interval)
[15:00:28] Reset PN for GTK
[15:00:28] f0:27:65:25:d5:f2: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 2 ARPs this interval)
[15:00:30] Reset PN for GTK
```



Test dispositivi – I° Dispositivo

cattura s4.pcapng

File Modifica Visualizza Val Cattura Analizza Statistiche Telefonia Wireless Strumenti Aiuto

Aplica un filtro di visualizzazione ...<Ctrl>/>

No.	Time	Source	Destination	Protocol	Length	Info
253	7.78478544	Azurewav_c4:db:69	(... 802.11		40	Acknowledgement, Flags=.....C
254	7.785122828	MurataMa_25:d5:f2	Azurewav_c4:db:69	802.11	54	Null function (No data), SN=2251, FN=0, Flags=.....TC
255	7.785498195	MurataMa_25:d5:f2	MurataMa_25:d5:f2	(f... 802.11	40	Acknowledgement, Flags=.....C
256	7.788992848	MurataMa_25:d5:f2	Azurewav_c4:db:69	802.11	54	Null function (No data), SN=2252, FN=0, Flags=.....TC
257	7.790134169	MurataMa_25:d5:f2	MurataMa_25:d5:f2	(f... 802.11	40	Acknowledgement, Flags=.....C
258	7.792982577	MurataMa_25:d5:f2	Azurewav_c4:db:69	802.11	179	QoS Data, SN=19, FN=0, Flags=p.....TC
259	7.792232382	MurataMa_25:d5:f2	MurataMa_25:d5:f2	(f... 802.11	40	Acknowledgement, Flags=.....C
260	7.792735777	MurataMa_25:d5:f2	Azurewav_c4:db:69	802.11	54	Null function (No data), SN=2253, FN=0, Flags=.....TC
261	7.793107452	MurataMa_25:d5:f2	MurataMa_25:d5:f2	(f... 802.11	40	Acknowledgement, Flags=.....C
262	7.802108517	MurataMa_25:d5:f2	ThuGmon_6f:dc:de:f2	802.11	111	QoS Data, SN=20, FN=0, Flags=.....TC

Frame Control Field: 0x8841
.000 0001 0011 1010 = Duration: 314 microseconds
Receiver address: Azurewav_c4:db:69 (48:5d:60:c4:db:69)
Destination address: Azurewav_c4:db:69 (48:5d:60:c4:db:69)
Transmitter address: MurataMa_25:d5:f2 (f0:27:65:25:d5:f2)
Source address: MurataMa_25:d5:f2 (f0:27:65:25:d5:f2)
BSS Id: Azurewav_c4:db:69 (48:5d:60:c4:db:69)
STA address: MurataMa_25:d5:f2 (f0:27:65:25:d5:f2)
.... 0000 = Fragment number: 0
0000 0001 0011 = Sequence number: 19
Frame check sequence: 0x6cc92fe9 [correct]
[FCS Status: Good]
Qos Control: 0x0000
CCMP parameters
CCMP Ext. Initialization Vector: 0x000000000001
Key Index: 0

Data (115 bytes)

```
0000 09 00 1a 00 2f 48 00 00 ef 8a 7d 10 00 00 00 00 .....H. ....)
0010 10 02 85 09 a9 00 09 01 00 00 88 41 3a 01 48 5d .....A: H)
0020 60 c4 db 69 f0 27 65 25 d5 f2 48 5d 60 c4 db 69 ..i.e%..H) .i
```

Frame (frame), 179 byte

```
[14:59:25] Reset PN for GTK
[14:59:25] f0:27:65:25:d5:f2: sending a new 4-way message 3 where the GTK has a zero RSC
[14:59:25] f0:27:65:25:d5:f2: DHCP reply 192.168.100.2 to f0:27:65:25:d5:f2
[14:59:27] Reset PN for GTK
[14:59:27] f0:27:65:25:d5:f2: sending a new 4-way message 3 where the GTK has a zero RSC
[14:59:27] f0:27:65:25:d5:f2: client has IP address -> now sending replayed broadcast ARP packets
[14:59:27] f0:27:65:25:d5:f2: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 0 ARPs this interval)
[14:59:27] f0:27:65:25:d5:f2: IV reuse detected (IV=1, seq=19). Client reinstalls the pairwise key in the 4-way handshake (this is bad)
[14:59:29] Reset PN for GTK
[14:59:29] f0:27:65:25:d5:f2: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 1 ARPs this interval)
[14:59:31] Reset PN for GTK
[14:59:31] f0:27:65:25:d5:f2: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 2 ARPs this interval)
[14:59:31] f0:27:65:25:d5:f2: sending broadcast ARP to 192.168.100.1 (sent 3 ARPs this interval)
[15:00:20] f0:27:65:25:d5:f2: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 2 ARPs this interval)
[15:00:22] Reset PN for GTK
[15:00:22] f0:27:65:25:d5:f2: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 3 ARPs this interval)
[15:00:24] Reset PN for GTK
[15:00:24] f0:27:65:25:d5:f2: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 4 ARPs this interval)
[15:00:26] Reset PN for GTK
[15:00:26] f0:27:65:25:d5:f2: Client DOESN'T reinstall the group key in the 4-way handshake (this is good)
[15:00:26] f0:27:65:25:d5:f2: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 1 ARPs this interval)
[15:00:28] Reset PN for GTK
[15:00:28] f0:27:65:25:d5:f2: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 2 ARPs this interval)
[15:00:30] Reset PN for GTK
```



Test dispositivi – II° Dispositivo

```
[15:45:09] Reset PN for GTK
[wlan0: STA 00:27:15:e7:e6:5a IEEE 802.11: authenticated
[wlan0: STA 00:27:15:e7:e6:5a IEEE 802.11: associated (aid 1)
[wlan0: AP-STA-CONNECTED 00:27:15:e7:e6:5a
[wlan0: STA 00:27:15:e7:e6:5a RADIOS: starting accounting session E1755FF0072FD1EB
[15:45:10] 00:27:15:e7:e6:5a: 4-way handshake completed (RSN)
[15:45:10] 00:27:15:e7:e6:5a: DHCP reply 192.168.100.2 to 00:27:15:e7:e6:5a
[15:45:10] 00:27:15:e7:e6:5a: DHCP reply 192.168.100.2 to 00:27:15:e7:e6:5a
[15:45:11] Reset PN for GTK
[15:45:11] 00:27:15:e7:e6:5a: sending a new 4-way message 3 where the GTK has a zero RSC
[15:45:11] 00:27:15:e7:e6:5a: client has IP address -> now sending replayed broadcast ARP packets
[15:45:11] 00:27:15:e7:e6:5a: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 0 ARPs this interval)
[15:45:11] 00:27:15:e7:e6:5a: received a new message 4
[15:45:13] Reset PN for GTK
[15:45:13] 00:27:15:e7:e6:5a: sending a new 4-way message 3 where the GTK has a zero RSC
[15:45:13] 00:27:15:e7:e6:5a: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 1 ARPs this interval)
[15:45:13] 00:27:15:e7:e6:5a: received a new message 4
[15:45:15] Reset PN for GTK
[15:45:15] 00:27:15:e7:e6:5a: sending a new 4-way message 3 where the GTK has a zero RSC
[15:45:15] 00:27:15:e7:e6:5a: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 2 ARPs this interval)
[15:45:15] 00:27:15:e7:e6:5a: received a new message 4
[15:45:17] Reset PN for GTK
[15:45:17] 00:27:15:e7:e6:5a: sending a new 4-way message 3 where the GTK has a zero RSC
[15:45:17] 00:27:15:e7:e6:5a: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 3 ARPs this interval)
[15:45:17] 00:27:15:e7:e6:5a: received a new message 4
[15:45:19] Reset PN for GTK
[15:45:19] 00:27:15:e7:e6:5a: sending a new 4-way message 3 where the GTK has a zero RSC
[15:45:19] 00:27:15:e7:e6:5a: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 4 ARPs this interval)
[15:45:19] 00:27:15:e7:e6:5a: received a new message 4
[15:45:21] Reset PN for GTK
[15:45:21] 00:27:15:e7:e6:5a: sending a new 4-way message 3 where the GTK has a zero RSC
[15:45:21] 00:27:15:e7:e6:5a: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 1 ARPs this interval)
[15:45:21] 00:27:15:e7:e6:5a: received a new message 4
[15:45:21] 00:27:15:e7:e6:5a: Client reinstalls the group key in the 4-way handshake (this is bad).
[15:45:21] 00:27:15:e7:e6:5a: OR client accepts replayed broadcast frames (see --replay-broadcast).
[15:45:23] Reset PN for GTK

[15:46:00] 00:27:15:e7:e6:5a: Received a new message 4
[15:46:08] Reset PN for GTK
[15:46:08] 00:27:15:e7:e6:5a: sending a new 4-way message 3 where the GTK has a zero RSC
[15:46:08] 00:27:15:e7:e6:5a: received a new message 4
[15:46:10] Reset PN for GTK
[15:46:10] 00:27:15:e7:e6:5a: sending a new 4-way message 3 where the GTK has a zero RSC
[15:46:10] 00:27:15:e7:e6:5a: received a new message 4
[15:46:10] 00:27:15:e7:e6:5a: client DOESN'T reinstall the pairwise key in the 4-way handshake (this is good) (used standard attack).
[15:46:12] Reset PN for GTK
[15:46:12] 00:27:15:e7:e6:5a: sending a new 4-way message 3 where the GTK has a zero RSC
[15:46:12] 00:27:15:e7:e6:5a: received a new message 4
```



Test dispositivi – III° Dispositivo

Non presenta vulnerabilità relative alla
reinstallazione della chiave di sessione
PTK, né GTK

```
[16:12:25] 8c:2d:aa:12:d4:9a: sending a new 4-way message 3 where the GTK has a zero RSC
[16:12:25] 8c:2d:aa:12:d4:9a: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 1 ARPs this interval)
[16:12:27] Reset PN for GTK
[16:12:27] 8c:2d:aa:12:d4:9a: sending a new 4-way message 3 where the GTK has a zero RSC
[16:12:27] 8c:2d:aa:12:d4:9a: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 2 ARPs this interval)
[16:12:29] Reset PN for GTK
[16:12:29] 8c:2d:aa:12:d4:9a: sending a new 4-way message 3 where the GTK has a zero RSC
[16:12:29] 8c:2d:aa:12:d4:9a: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 3 ARPs this interval)
[16:12:31] Reset PN for GTK
[16:12:31] 8c:2d:aa:12:d4:9a: sending a new 4-way message 3 where the GTK has a zero RSC
[16:12:31] 8c:2d:aa:12:d4:9a: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 4 ARPs this interval)
[16:12:32] 8c:2d:aa:12:d4:9a: Client DOESN'T reinstall the pairwise key in the 4-way handshake (this is good) (used standard attack).
[16:12:33] Reset PN for GTK
[16:12:33] 8c:2d:aa:12:d4:9a: sending a new 4-way message 3 where the GTK has a zero RSC
[16:12:33] 8c:2d:aa:12:d4:9a: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 1 ARPs this interval)
[16:12:35] Reset PN for GTK
[16:12:35] 8c:2d:aa:12:d4:9a: sending a new 4-way message 3 where the GTK has a zero RSC
[16:12:35] 8c:2d:aa:12:d4:9a: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 2 ARPs this interval)
[16:12:37] Reset PN for GTK
[16:12:37] 8c:2d:aa:12:d4:9a: sending a new 4-way message 3 where the GTK has a zero RSC
```

```
[16:12:45] Reset PN for GTK
[16:12:45] 8c:2d:aa:12:d4:9a: sending a new 4-way message 3 where the GTK has a zero RSC
[16:12:45] 8c:2d:aa:12:d4:9a: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 3 ARPs this interval)
[16:12:47] Reset PN for GTK
[16:12:47] 8c:2d:aa:12:d4:9a: sending a new 4-way message 3 where the GTK has a zero RSC
[16:12:47] 8c:2d:aa:12:d4:9a: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 4 ARPs this interval)
[16:12:49] Reset PN for GTK
[16:12:49] 8c:2d:aa:12:d4:9a: sending a new 4-way message 3 where the GTK has a zero RSC
[16:12:49] 8c:2d:aa:12:d4:9a: Client DOESN'T reinstall the group key in the 4-way handshake (this is good)
[16:12:49] 8c:2d:aa:12:d4:9a: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 1 ARPs this interval)
[16:12:52] Reset PN for GTK
[16:12:52] 8c:2d:aa:12:d4:9a: sending a new 4-way message 3 where the GTK has a zero RSC
[16:12:52] 8c:2d:aa:12:d4:9a: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 2 ARPs this interval)
[16:12:54] Reset PN for GTK
[16:12:54] 8c:2d:aa:12:d4:9a: sending a new 4-way message 3 where the GTK has a zero RSC
[16:12:54] 8c:2d:aa:12:d4:9a: sending broadcast ARP to 192.168.100.2 from 192.168.100.1 (sent 3 ARPs this interval)
```



Conclusioni

Attacco abbastanza grave, ma comunque limitato all'area intorno al target.

Buona norma avvalersi:

- di protocolli sicuri come HTTPS, SSL/TLS, SSH
- delle reti VPN

Unico modo per proteggersi è aggiornare i sistemi con le ultime patch messe a disposizione.

Cambiare password e renderle più complesse non serve a mitigare l'attacco.

Sviluppi futuri

- Testare più dispositivi
- Utilizzare gli altri script messi a disposizione da Mathy Vanhoef
- Approfondire l'analisi dei pacchetti su Wireshark

**Scuola Politecnica e
delle Scienze di Base**



Università degli Studi di Napoli Federico II

Facoltà di Ingegneria delle Telecomunicazioni

FINE

Network Security - Bianca Ruggiero M59/167